

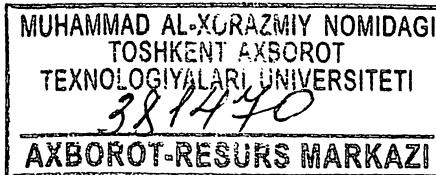
Жавлон АБДУЛЛО

МУКАММАЛ ДАСТУРЛАШ

1-китоб

HTML ва CSS

Китоб касб-хунар коллекци, академик лицей ва
юқори синф ўқувчилари, олий ўқув юртлари
талабалари ҳамда дастурлашни ўрганмоқчи бўлган
кенг китобхонлар оммаси учун мўлжалланган



**ТОШКЕНТ
"MUSIQA"
• 2017**

УЎК: 004.42

КБК:32.973.202-018

A15

АБДУЛЛО, Жавлон

Мукаммал дастурлаш. (1-китоб: HTML ва CSS) Жавлон АБДУЛЛО
– Тошкент, «Musiqा» нашриёти, 2017 / 224 б.

Ушбу китоб замонавий дастурлашни ўрганмоқчи бўлганлар, дастурлаш билан шуғуланадиганлар ва ҳатто дастурий таъминот яратиш бўйича етук жутахассислар учун ҳам мўлжалланган бўлиб, илм олиш ва мустаҳкамлаш учун энг керакли бўладиган зарурӣ билимларни ўзида жамлаган ҳамда содда, тушунарли тилда баён этилган, ортиқча маълумотлар ўрнига ўн йилдан зиёд амалий ишлаш жараёнида орттирилган ҳаётий тажрибалардан келиб чиққан ҳолда фойдали кўрсатмалар самарали тарзда ва тўғри келтирилган мукаммал қўлланмаларнинг илк қисмидир. Шунинг ўзи ҳам бундаги ёритилган билимларни ўзлаштирган кишининг етук веб-дизайнер сифатида кенг фаолият юритишига етарли замин яратади.

Китобда ёритилган мавзулар бўйича мулоҳазалар, таклифлар, саволлар ҳамда ўз амалий дастурлаш фаолиятингизда юзага келган муаммоларни ilm.yurt.uz сайти орқали йўллашингиж мумкин.

УЎК: 004.42

КБК:32.973.202-018

Ушбу китобни ва унинг саҳифаларини муаллиф руҳсатисиз ҳар қандай кўринишида кўпайтишиш ва тарқатиши қонунан таъқиб этилади.

ISBN:978-9943-4780-7-8

© Жавлон АБДУЛЛО,

© «Musiqা», 2017

I БЎЛИМ

КИРИШ



1.1. Китоб ҳусусида

Ҳозирда шиддат билан ривожланиб бораётган ҳаётимизни интернетсиз тасавур қила олмаймиз. Ишлаб чиқарган маҳсулотларимизни, илмий ва адабий меросимизни, юксак маданиятимизни кенг оммага, ҳатто бутун жаҳонга ёйишимизда глобал тармоқнинг bemisл имкониятларини четлаб ўтолмаймиз.

Интернетда маълумотлар қандай алмашинишни, ўқувчига қай тарзда намоён бўлишини дастурловчи назарида разм солиш учун эса аввало HTML ва CSS тушунчаларидан хабардор бўлиш лозим. Аслида булар дастурлаш тиллари сифатида қаралмайди ва фақат веб саҳифаларни акслантиришда фойдаланилади холос. Сайтлар ва веб-саҳифаларнинг бежирим, кулагай интерфейсда ҳамда фойдаланувчига тушунарли, шу билан бирга оммабоп браузерларда кутилганда гидек намоён бўлиши учун HTML ва CSS имкониятларни пухта ўрганиб, улардан ўз ўрнида тўғри фойдаланадиган бўлишимиз керак. Бу йўлда ушбу китоб доимий, шу билан бирга самарали фойда берувчи ҳамроҳ бўлади.

Ушбу мукаммал қўлланма дастурлаш тилини ўрганишга энди қадам қўймоқчи бўлган ўқувчига ҳам, бир неча ийиллар мобайнида дастурлаш билан шуғулланадиганлар учун ҳам, дастурлашни мукаммал ўзлаштириб олганлар учун ҳам мўлжалланган бўлиб, ўз билимларини мустаҳкамлаш йўлида улкан ҳисса қўшишига ишонализ. Зоро бунда янги технологик даражага кўтарилиган

ва олдингиларидан жуда катта тафовутга эга HTMLнинг бешинчи ва CSSнинг учинчи талқини асос сифатида қаралиб, HTML ва CSS имкониятлари батафсил ёритилган. Ҳозирги кунда қўл соатидан тортиб мобил телефонлару планшетлар, компьютерлару маҳсус серверлар, ҳатто телевизорлару «ақлли технология» номини олган барча ускуналар HTML ва CSSни қўлловчи ўз дастурий таъминотларига эга. Ушбу имконият уларда акслантириш мумкин бўлган барча нарсаларни намоён қилиш учун етарли, аммо технология тараққиёти бир жойда тўхтаб қолмайди. Кейинчалик киритиладиган муҳим янгиликларни маскур китобда кўрсатилган сайтда ва қўлланманинг кейинги нашрларида қўшиб борилади. Ифода этилган кодлар қандай тавсифланган бўлса, барча оммабоп браузерлар, яъни Mozilla Firefox, Google Chrome, Microsoft Edge, Opera, Safari, Internet Explorer ва бошқаларнинг юқори талқинларида, ҳамда мобил браузерларда бирдек намоён бўлади. Маҳсус истесънолар ўз ўрнида баён этилган. Катта ҳажмли ва қўп имкониятли сайtlар яратиш учун ҳам маҳсус кўрсатмалар бериб борилган.

1.2. Қисқача тарих

1986 йил стандартлаштириш бўйича халқаро ташкилот (ISO) Standard Generalized Markup Language (SGML) – «Стандарт Умумлаштирилган жойлаштириш тили стандарти» номли стандартни қабул қилган. У ISO 8879 рақами бўйича рўйхатдан ўтган бўлиб, турли операцион тизимлар ва киритиш қурилмаларида ҳам қулай ўзлаштириладиган матн шаклидаги маълумотларни акслантиришга мўлжалланган. Унда маълумотлар қандай кўринишда ифодаланиши келтирилган холос, маҳсус тил сифатида барча буйруқлар жамланмаган ва уларнинг айнан қай тарзда намоён бўлиши батафсил келтирилмаган. Шунинг учун ҳам ушбу стандарт оммалашмай қолган.

1991 йил Женевадаги Европа заррачалар институ-

ти (CERN)нинг британиялик олими Тим Бернерс Ли маълумотларни тўғри саҳифалашни билмайдиганлар учун ҳам электрон тарзда хужжатларни алмашишни қулаштириш мақсасида маҳсус тил яратди. Унга асос сифатида SGML олинган бўлиб, HTML – Hyper Text Markup Language (Ишоратларни жойлаштириш тили) деб номланди. Орадан икки йил ўтиб, 1993 йил қирқта белгиланиш – «тег» ни ўз ичига олган HTMLнинг 1.2-талқини яратилди ва ўша пайтдаги ягона график имкониятли браузер – «Mosaic» дастури ёрдамида маълумотлар акслантирилди. «Mosaic» NCSA – National Center for Supercomputer Applications – «Суперкомпьютер дастурлари учун миллий Марказ» маҳсулоти эди. Айнан шу марказда WWW (World Wide Web) – «умумжаҳон тўри» ҳам ишлаб чиқилган.

1994 йил «умумжаҳон тўри мутаносиблаштирувчиси» W3C (World Wide Web Consortium) ташкил этилиб, мавжуд тилга улкан ўзгартиришлар киритиб, 1995 йили HTMLнинг 2.0 талқини ҳавола этилди. Асосий ўзгариш, маълумотларни формада жамлаб серверга юбориш механизми қўшилганлигида бўлди. Аммо бу умуман талаб доирасида эмасди. Унга мураккаб математик формулялар, турли шаклдаги жадваллар, шу билан бирга расм билан матнни биргаликда акслантириш имкониятлари киритилиши лозим эди. Бироқ бундай имкониятларга эга браузернинг ўзи у пайтда мавжуд бўлмаган. Кенгайиб бораётган технологик имкониятларни инобатга олган ҳолда HTML муаллифлари 3-талқинни яратиш мобайнида янгича ёндашиш билан CSS воситасини ҳам амалга тадбиқ этишди. Яъни HTML орқали жойлаштириладиган объекtlарнинг кетма-кетлиги аниқланади холос, уларнинг қай тарзда намоён бўлиши эса алоҳида синтаксисга эга бўлган CSS ёрдамида кўрсатилади.

1994 йили «Mosaic» браузери яратувчилари Жеймс Кларк бошчилигида анчагина мукаммал «Netscape» номли янги браузер яратдилар ва илқ бор савдога чиқаришди. Унда амалиётга тадбиқ қилинган ва бир қанча тўлдиришлар киритилган HTMLнинг 3.2-талқини ва CSSнинг 1-талқини бекаму кўст намоён бўларди. Ҳатто бир неча

йилардан кейин ҳам Microsoft ширкати томонидан яратылган «Internet Explorer» браузерининг 3-талқинида ҳам, «Netscape Navigator»нинг 1-талқинида күп имконияттар киритилмаган эди. Инсонияттинг маълумот алмашишга ташналиги, ҳамда «Netscape Navigator»нинг изланишларга ўчлиги ва янгиликларни тезкор тадбиқ этиши, унинг 1995 йиллари дунёдаги энг тез ривожланаётган ширкат мақомини олишига сабабчи бўлган. Кейинчалик Microsoft ширкати W3C нинг ҳомийсига айланиб, ўз операцион тизимиға текин равища «Internet Explorer»ни қўшиб бериш билан аста-секин «Netscape Navigator»ни буткул ортда қолдирди. Шу боисдан ҳозирда энг оммалашган браузерлар сафида «Netscape Navigator» ўрин олмай қўйган, унинг яратувчилари эса такомиллаштириш устидаги фаолиятларини тўхтатишган.

1997 йили HTMLнинг 4-талқини жорий этилди. Унда бир қанча белгиланишлар эскирган деб топилиб, қўлламасликка ундей бошланди. Уларнинг ўрнига CSSнинг имкониятларидан фойдаланиш тавсия этилди. 1999 йили HTML 4.01-талқини стандарт асосида фойдаланишга чиқарилди. Сўнгра ўн йилдан зиёд вақт ушбу тилни такомиллаштириш буткул тўхтаб қолди. Сабаби яратувчилари HTMLнинг ўрнига XML (кенгайтирилган жойлаштириш тили)дан фойдаланишни маъқул топишди. Лекин браузер яратувчилари ушбу қарашни маъқуллашмади ва ўзлари талаб доирасида HTML ва CSS имкониятларини кенгайтириб бораверишди. Натижада битта қоидани турли браузерлар учун қайта-қайта ёзишга тўғри келадиган бўлиб қолди. HTML воз кечиб бўлмайдиган тил эканлигини англашганларидан сўнг, ушбу тилга жуда жиддий ёндашиш билан қаралиб, ўзаро келишмай, ягона стандартга қатъий риоя қилмай, тарқоқ ишлайдиган энг машхур ширкатлар (*Microsoft, Opera, Mozilla, Google, IBM, Apple* ва бошқалар) бирлашиб, янги HTML5 устидаги изланиш олиб боришга қарор қилишди. Ҳозирда 2013 йил февралда тақдим этилган HTML 5.0 талқини бўйича барча браузерлар фаолият олиб боришмоқда, унда овозли ва ҳаракатли тасвиirlарни ҳамда ўзга-

рувчан векторли график (SVG)ларни акслантиришга катта эътибор қаратилган. HTML5 ни ривожлантириш 2022 йилгача давом этиши мумкин, лекин орадан бир қанча йил ўтса ҳам унинг қонуниятларига сезиларли ўзгартириш киритилмади ва ушбу китобда ёритилган билимлар HTML5 ва CSS3 имкониятларини тўла қамраб ола олади.

HTML тилида хатоликлар ва тушунмовчиликлар эътиборсиз қолдирилади. Бу эса ҳамма дастурловчига ўзи истагандай код ёзиш имкониятини бериш билан бирга, ўзи шундоқ ҳам тартибсиз интернетни янада бесаранжомлашишига олиб келади, яъни айни бир маълумот ҳар хил браузерда турли шаклда намоён бўлади, изловчи тизимларда эса кутилган хабар топилишига тўскинлик вужудга келади. Бундай муаммоларни оддини олиш мақсадида, 2000 йилдан кенгайтирилган HTML тили (XHTML) яратилган. Унда анча қатъий талаблар қўйилган бўлиб, хатолик учраган ҳолда кутилган натижага эришилмайди. Бундай қатъийлик дастурий таъминот ёрдамида кодларни таҳдил қилишга қулайлик яратади. Афсуски, HTML5нинг талабларини қатъийлартиришга муҳим эътибор қаратилгандан сўнг XHTMLни қўллаш ва такомиллаштириш тўхтатилди.

II БЎЛИМ

HTMLНИНГ АСОСИЙ ТАРКИБИ



2.1. Асосий тушунчалар

HTML дастурлаш тили бўлмагани учун ҳам унинг кодларини бўйруқ эмас, белгиланиш сифатида қаралади. Ҳар бир белгиланиш «<» белгиси билан бошланиб, «>» белгиси билан якунланади ва уни «тег» деб номланади.

Ушбу китобда тег ва бошқа кодларнинг синтаксиси ортранги алоҳида ажраб турувчи маҳсус майдон ичидаги келтирилади. Уларни баён этишда тагига чизилган эгри ёзувдаги ўзбекча изоҳ ёзилған бўлса, унинг ўрнида ифода этилган код келтирилиши инобатга олинган бўлади. Агар қийматлар «|» белгиси билан ажратилган ҳолда берилса, келтирилган рўйхатдагидан фақат биттаси қўлланилишини англатади. Агар «[» ва «]» қавслари ичидаги ифода келган бўлса, у ёзилиши шарт бўлмаган кодлигини билдиради. Ёзилиши шарт бўлмаган ифодалар (тег ёки унинг аломати) ўзининг одатий қийматига эга бўлиб, ихтиёрий браузер ўша қиймат бўйича акслантираверади.

Синтаксисдан бошқа жойларда – китобнинг одатий ёзувлари ичидаги, «[2.1.]» каби белгиланиш учраса, баён этилган фикрнинг тўлдирувчи маълумоти ушбу китобнинг 2.1-бўлимида (яъни мос рақамлар остидаги мавзуда) кенгроқ ёритилганлигини англатади. Бирор тушунчани тафсилоти келтирилаётганда «*» қўйилған бўлса, демак унинг тўла изоҳи ушбу китобдаги бошқа мавзуда келтирилганлигидан далолатдир. Уни батафсилроқ ўрганиш керак бўлса, ўша сўзни китоб охиридаги атамалар

күрсатгичидан нечанчи саҳифада кенг ёритилғанлигини топиб, ўқиши мумкин.

Шу ўринда ушбу китобда қўлланиладиган баъзи тушунчаларга ойдинлик киритиб ўтайдик:

Браузер – Жойлаштириш тилларида (HTML, xHTML, XML ва бошқалар) ёзилган кодларни мазмунни бўйича намоён этувчи дастурний таъминот.

Ишорат – ҳужжатнинг умумжаҳон тўридаги манзилини аниқловчи ёзув. У кўп ҳолларда URL, URN, URI кўринишларида ҳам номланади. [2.3.]

Элемент – тегнинг браузерда аксланадиган кўриниши. Тег «<» ва «>» белгиларни орасидаги ёзувни ифодаласа, элемент деганда ёзув эмас, браузерда намоён бўладиган шакл тушунилади. Элемент браузерда тўғри тўртбурчак кўринишидаги ёки қаторли ёзув сифатидаги майдонни эгаллайди, бу майдон кўринувчи ёки кўринмас бўлиши мумкин. [2.5.]

Объект – Ягона ёки бир нечта элементлар мажмуасидан ҳосил бўлган шакл.

Веб-саҳифа – браузернинг битта ойнасида ифодаланадиган барча объектларнинг мажмуи. Келгусида уни оддий «саҳифа» деб ҳам юритамиз.

Лотин ҳарфлари – «A»дан «Z»гача ва «a»дан «z»гача ҳамда «_» белгисидан иборат жами 53 та белги.

Ажратувчи белгилар – ҳарфни ифодаламайди, одатда сўзларни бир биридан ажратиш учун қўлланилади. Улар «», «.», « », «(», «)», «+», «-», «/», «*» ва ҳоказо.

Идентификатор – элементларни бир-биридан фарқлаш учун қўлланиладиган номланиш. Унда фақат лотин ҳарфлари, пастки чизиқча () ва ракамлардан фойдаланилади, ажратувчи белгилар ишлатилмайди. Бундан ташқари идентификаторнинг биринчи белгиси рақамдан иборат бўлмаслиги шарт. Идентификаторда катта-кичик ҳарфлар аҳамиятга олинади, ҳамда уни номлашда тилда мавжуд бўлган «захираланган қалит сўзлар», яъни буйруқ ёки бошқа маънони англатувчи сўзлар қўлланмаслиги керак. Фақат HTML тилидагина идентификаторни қўштироқ ичида бериш имконияти мавжудлиги учун ҳам баъзи ажратувчи белгилардан

фойдаланиш мумкин. Бундай белгиланишларни «**келишілган идентификатор**» деб номлаймиз, унда бириңчи ҳарфдан кейин чизиқча (-) құлланилиши мумкин. Аммо биз келажақда бошқа дастурлаш тилларини ҳам үзлаштиришимиз зарурлигини инобаттан олған ҳолда, идентификаторда умуман ажратувчи белгилардан фойдаланишни тавсия этмаймиз. HTML дастурлаш тили бўлмагани учун ҳеч қандай математик амаллар код ичидага ёзилган деб қаралмайди, акс ҳолда чизиқча айриш амали сифатида қабул қилиниши мумкин бўларди.

Юқоридаги келишувларга асосланиб, тегнинг мавжуд икки хил синтаксисини келтирамиз:

`<tag номи [tag аломатлари]>`

`<tag номи [tag аломатлари]>[tag асоси]</tag номи>`

Юқорида таъкидлаб ўтканимиздай, айрим веб саҳифалар дастурий таъминотлар ёрдамида ўқилиб, ундағы маълумотларни үзлаштириш қулай бўлиши учун XHTML стандартлари бўйича ёзилади. Унда бириңчи синтаксис бўйича ёзилган тег хато ҳисобланади. Албатта ёпувчи тег номи кўрсатилиши керак, агар у мавжуд бўлмаса, қуидагича ифодаланади:

`<tag номи [tag аломатлари] />`

XHTMLни ривожлаштириш устида фаолият тўхтатилганлигини ва ушбу китоб айнан HTMLни ёритишига асослангани учун, биз тегларни HTML5 талаблари бўйича ёзишни тавсия этамиз.

Шу ўринда кичик бир мисол қўриб ўтгайлик. HTMLда сатрларни бўлувчи тег – BR, у ёпувчи қисмга эга эмас, шунчаки `
` шаклида қўлланилади. Демак, юқоридаги фикрни инобатта олған ҳолда уни `
` ё `
` ёки `
` кўринишларида ёзиш мумкин. Ҳар учаласи ҳам тўғри ҳисобланади ва бир хил натижга беради. Биз оддийгина `
` шаклида ёзишни тавсия этамиз, бу HTMLни асл синтаксисига мос келиш билан бирга, ёзилаётган код ҳажмини қисқартишига ҳам ёрдам беради.

HTML тилида ҳарфларнинг катта кичиклигига эътибор қаратилмайди. Юқоридаги тегни **
**, **
**, **
** ва **
** шакллари бирдек қабул қилинади. Бошқа дастурлаш тилларида (Паскал бундан мустасно) фақат кичик ҳарфларни қўллаш талаб қилингани учун, дастурловчилар орасида кичик ҳарфлардан фойдаланиш оммалашган. Тег номларини қандай ёзиш дастурловчига ҳавола, аммо биз ҳам кичик ҳарфлардан фойдаланишини мақсадга кўпроқ мувофиқ деб ҳисоблаймиз.

Тегнинг аломатлари қуийдаги қўринишида ёзилади:

аломат номи = қиймат

Қиймат одатда қўштириноқ ичида ёзилади, унда ажратувчи белгилар, ҳусусан бўш жой (пробел) қатнашмаган бўлса, қўштириноқ ёзиш шарт эмас. HTML тилида ҳам қўпгина дастурлаш тиллари сингари икки хил қўштириноқ ишлатилади: «'» ва «"». Аммо кўтгчилик дастурловчиларга одат тусига киргanganligini учун, иккиталик қўштириноқни доим қўллаш тавсия этилади. Китобнинг кейинги қисмларида синтаксис шу тариқа берилади, дастурчининг ўзи ихтиёрига қараб истаган қўштириноғини қўйиши ёки улардан умуман фойдаланмаслиги мумкин.

Матн ичида ёзиладиган баъзи маҳсус кодланишларни келтириб ўтамиш:

Белгиси	Коди	Қўриниш	Изоҳ
<	<	<	Кичик белгиси
>	>	>	Катта белгиси
"	"	"	Иккиталик қўштириноқ
	'	'	Битталик қўштириноқ (опостраф)
‘	‘	'	Чап битталик қўштириноқ
’	’	'	Ўнг битталик қўштириноқ
&	&	&	Амперсанҷ, мантикий қўпайтириш ишораси
 	 		Ажратмас бўш жой

Белгиси	Коди	Қўриниш	Изоҳ
	‒	-	Чизиқча
	‐	-	Қисқа чизиқча
−		-	Математик айириш белгиси
–	–	-	Қисқа чизик, «п» ҳарфи узунлигига
—	—	-	Ўртача чизик, «т» ҳарфи узунлигига
	―	-	Узун чизик
­			Бўғин кўчирувчи фақат қатор охирида чизиқча бўлиб қўринади

Барча маҳсус кодланишларни келтириб ўтмаймиз, қолган белгилар ўз қўринишига эга ва уни ҳужжат ичидаганда ёзилса, кодлама тўғри танланганда, барча браузерлар айнан ўзидек акслантирилади. Юқорида чалкашликларга сабабчи бўлувчи белгиланишлар келтирилган, агар уларни чоп этишга зарурият туғилса, жадвалда келтирилган «Белгиси» ёки «Коди»дан фойдаланиш тавсия этилади.

Чоп этиладиган матн катта бўлган ҳолларда, улардаги сўзларнинг эгаллайдиган ҳажмига қараб, бир неча сатрларга бўлинниб акслантирилади. Бундай ҳолларда ёнмаён турмаса мазмуни ўзгариб кетадиган сўзлар кетма-кет қаторларга тушиб, жумла моҳиятини бузиши мумкин. Шундай англашилмовчиликка йўл қўймаслик учун «ажратмас бўш жой» ишлатилади, у маҳсус ибораларни англатувчи кўшни сўзларни турли қаторларга тушиб қолмаслигига хизмат қиласди ва бир қаторда ёнмаён келиши шарт бўлган сўзлар орасига қўйилади. Браузерлар эса ушбу сўзларни ажратилган ҳолда намоён этаверади. Аксинча жуда узун сўзларнинг маълум бир қисмини бошқа сатрга кўчириб туширмаса, саҳифанинг чироий бузиладиган бўлса, «­» белгисидан фойдаланилади. Агар браузер бўғин кўчиришга ҳожат йўқ деб ҳисобласа, уни сўздаги ҳарфлар орасида кўрсатмайди; кейинги қаторга сўзларнинг ажратилган қисмини туширишга тўғри келса, қаторнинг охирида уни чизиқчага алмаштиради.

HTML тилида бўш жойлар, табуляция ва қатор якуни-ни билдирувчи белгилар нечта ва қандай кетма-кетлиқда ёзилган бўлса ҳам, ягона бўш жой сифатида акслантирилади. Матн қандай кўринишида ёзилган бўлса, айнан шундай намоён бўлиши учун `pre` [8.8] ёки `xmp` [3.3.] теги ичида келтириш талаб қилинади. Сабаби HTML – жойлаштириш тили, матннинг қайси бўлаги қандай кўринишида чоп этилиши керак бўлса, уларни шунга мос теглар ичида ёзиш лозим.

Барча тилларда бўлгани каби, HTMLда ҳам коднинг браузер таҳлил қиласлик керак бўлган қисмини ажратиб қўйиш мумкин. Бундай бўлак – изоҳ деб номланади ва қўйидаги тег ичида ёзилади:

```
<!-- изоҳ матни -->
```

2.2. Илк босқич

Веб саҳифа яратиш учун биз оддий матн таҳрирловчи дастурдан фойдаланамиз. Аслида турли-туман HTML яратувчи кулай муҳитлар сероб, аммо биз ўрганувчи бўлган лигимиз ва ҳар бир кодни ўзимиз нима учун ёзаётганилигимизни англаб етишимиз учун «`notepad`» ишлатишимииз мақсадга мувофиқ. Ёзишни осонлаштириш ва кодларни ранглар билан фарқлаб кўриш учун «`notepad++`»ни кўлласа ҳам бўлади. Бироқ, бошланғич қадамда HTML кодларни ўзи ёзиб берадиган муҳитлардан фойдаланилмагани маъқул, сабаби ўзи қўшиб қўядиган кодлар нимани англатишини ҳали тўлиқ ўрганиб чиқмадик.

Биринчи қадамда матн таҳрирловчи дастур (`notepad`, `notepad++` ёки `boishqa`, бундан кейин МТД деб юритамииз) ёрдамида янги файл яратамииз. Уни «UTF-8» шаклида кодламага ўтказиб оламиз, сабаби бутун дунёда мавжуд белгиларнинг ҳаммаси унда жамланган, ҳамда у айнан веб-саҳифалар яратишга йўналтирилган.

Иккинчи қадамда файлга ном бериб, кенгайтмасини «`htm`» (ёки «`html`», кўпгина кенгайтмалар учта ҳарфдан иборат бўлгани учун ҳам «`htm`»ни танлаш тавсия этила-

ди) деб сақлаш керак. Кейинги қадамларда бевосита код ёзиб борилаверилади.

5-талқин тадбиқ этилмасдан аввал HTMLда ресурсларнинг етишмаслиги ва аниқ ягона келишувга амал қилинмаслиги код ёзишда жуда қўп музокарали қарашларни келтириб чиқарар эди. Ёзилган кодни ягона қолипга солиш учун жуда мураккаб ва эсда қолиши мушкул бўлган синтаксис асосида хужжат турини кўрсатиш керак бўлган. Агар саҳифанинг таркиби кескин ўзгартирилса, унга мос равищда хужжат тури ҳам ўзгартирилиши тақазо этилган. Хужжат турини аниқловчи DTD файл W3C сайтидан олинган. Агарда сайт билан тўғридан-тўғри алоқа мавжуд бўлмаган жойда ишлаш учун веб-саҳифа ёзилиши керак бўлса, яна муаммо очиқлигича қолаверган. Ҳозирда барча браузерларнинг юқори талқинлари «HTML5»ни қўллашини инобатга олсан, веб-саҳифани DOCTYPEнинг соддагина кўринилишидан бошлийверамиз. Демак, илк қатор доим қўйидагича бўлади:

```
<!DOCTYPE html>
```

Буни ва таркибни аниқловчи бошқа тегларни ёзилмаса ҳам браузерлар уларни бор деб ҳисоблайверишиди. Аммо биз ёзаётган кодимизга «ишласа бўлди» деган нуқтаи назар билан қарамаймиз, тўғри ва талаб бўйича бехато код ёзишни ўрганамиз. Зоро биз ёзган ихтиёрий кодни таҳлил қилувчи ташки дастур ёки ўзига индексация қилувчи (маълум қолипда кўчириб олевчি) излаш тизимлари (масалан *Google*, *Bing*, *Yandex*, *Rambler* ва ҳоказолар) ҳам нима ёзганимизни тўғри ва тўлиқ тушунсин.

HTMLнинг кўпгина теглари ҳозирда эскирган деб қаралади ва фойдаланиш тавсия этилмайди. Аммо ўз вазифасини ҳали ҳам бажараверади. Келгусида бундай «эскилил сарқитлари»дан ҳам бутунлай воз кечиши тақозо этилади. Сабаби бир неча йиллардан кейин янги браузерлар бундай кодларни акслантирумай қўйишлари ҳам мумкин.

Яна бир муҳим жиҳат, ҳар бир элементнинг пайдо

бўлиш жойи ва шаклига мос равищда тегларни ўз ўрнида ёзишга катта аҳамият бериш лозим, акс ҳолда турли чалкашликлар келиб чиқиб, кейинчалик камчиликни бартараф этиш мушкуллашиши мумкин.

2.3. Асосий қисмлар

HTML ишоратлар билан ишловчи тил бўлгани учун ҳам аввало, ишорат ўзи нима ва қандай кўринишга эгалигини аниқлаб олишимиз керак. Инглиз тилида уни URL – «Uniform (аввал унинг ўрнида Universal сўзи ишлатилган) Resource Locator» деб номлаш оммалашган, яъни манбаънинг ягона (универсал) кўрсатгичи. Бу глобал тармоқ – интернетда, жойлашган манъбанинг манзилини аниқловчи ишорат. Унинг умумий кўриниши куйидаги шаклда бўлади:

услуб://таҳаллус:калитсўз@хост:порт/йўл?омиллар#бўлак

Бунда белгиланган сўзлар куйидаги манъоларни англатади:

- **услуб** – веб-манбага мурожаат қилиш услуби, қўп ҳолларда тармоқ қайдномаси назарда тутилади, масалан http, ftp, https, telnet, mailto, file, data, news ва ҳоказо.
- **таҳаллус** – манбага кириш имконияти бўлган фойдаланувчи таҳаллуси.
- **калитсўз** – фойдаланувчининг маҳфий кириш сўзи. Шунинг учун ҳам баъзида калит сўзларда «@» белгисини ишлатиш тавсия этилмайди.
- **хост** – домен (сайт, сервер компьютер) номи ёки унинг IP манзили.
- **порт** – уланиш амалга ошириладиган порт рақами.
- **йўл** – манъбанинг хостда жойлашган манзилини аниқлаштирувчи йўл. Бунда аниқ сервердаги имкон берилган папка манзили ёки қийматлар кетма-кетлиги «/» билан ажратилган ҳолда кўрсатилиши мумкин. Сервер ўзининг созланишига қараб, қийматларни мос тарзда қабул қиласди.
- **омиллар** – саҳифани кўриш учун қўлланиладиган,

қўшимча киритилиши керак бўлган қийматлар, улар «&» белгиси билан ажратилган ҳолда ўзгарувчи номига берилади.

▪ **бўлак** – саҳифа узун бўлган ҳолларда айнан қайси бўлак экран бошида кўринишини таъминловчи белгиланиш.

Агар манба ҳамма учун очиқ бўлса, унда таҳаллус ва калит сўзлар ёзилмайди. Порт рақами кўрсатилмаган ҳолларда уни қайдноманинг одатий қийматида берилгандай қабул қиласди. Ишоратга мисол кўриб ўтамиз:

```
http://yurt.uz/?a=16
```

Омилларда кўрсатилган «а»нинг қийматини ўзгартирилса, саҳифа ҳам мос равишда ўзгаради.

HTML учта асосий қисмдан ташкил топади.

Биринчиси – DOCTYPE. У ҳақида олдинги мавзуда тўхталиб ўтдик. [2.2.] У ҳужжат XML, XHTML ёки бутунлай бошқа тилда эмас, айнан HTMLда ёзилганлигини англатади. Ҳужжатни ўқиётган браузер ёки бошқа дастур айнан қандай келишув асосида таҳлил қилиш кераклигини англашиб олиши учун бу жуда муҳим.

Иккинчиси – бош қисми. У **head** теги ичидаги ёзилади ва ҳужжатни шакллантириш учун керакли маълумотларни ўзида жамлайди. Унда ҳужжат қандай кодламада эканлиги, нишончаси, сарлавҳаси ҳамда унга қўшимча юкланидиган файллар рўйхати ва бошқа муҳим маълумотлар ёзилади. Ҳар бир қўшимча кўрсатилган файл юкланиб бўлиши билан ишга тушади, яъни ундаги кодлар таҳлил қилиниб акслантирилади. Шунинг учун ҳам бош қисмидаги браузерда кўринувчи ифодалар ёзиб бўлмайди дея олмаймиз. Аммо бундай ёзиш тавсия этилмайди.

Учинчиси – асосий қисм ёки тана қисми. У **body** тегининг ичидаги ёзилади ва браузернинг акслантирувчи майдонини тўла эгаллайди. Барча кўринувчи (ва ҳатто кўринмайдиган) тасвиirlар ундан ўрин олади. Шу ўринда аниқлик киритиб ўтиш керакки, ҳозирда **head** ва **body** тегларининг кескин фарқи қолмаган. Бирида ёзилиши

керак бўлган тегни бошқасида ёзса ҳам, ҳеч хатосиз бажарилаверилади. Бу бир тарафдан жуда чалкашликка олиб келса-да, иккинчи тарафдан керакли файларни фақат унга мурожаат бўлгандагина юклаб олиб бажаришни таъминлайди ва шу билан тезкор маълумот алмашиниб, сайтларни тез ишилашига замин яратади. Маъдомики шундай экан, биз head ва body орасида мантиқий чегара ясаб, бош қисмни фақат хужжатни шакллантиришда керак бўладиган тавсиф-хабар¹лар йиғадиган бўлим; асосий қисмни эса браузерда кўриниш ва маълумот акслантириш учун хизмат қиласидан бўлим сифатида қараймиз.

Кейинчалик киритилган DOCTYPEни инобатта олмаганда барча теглар, аниқроғи иккинчи ва учинчи қисмлар ягона html теги ичида ёзилади. DOCTYPE, HTML, HEAD, BODY – бу коднинг бўлимларини билдириб, битта веб-саҳифа учун фақат бир марта ёзилади.

2.4. Саҳифанинг бош қисми

Келишиб олган мантиқий чегарамизга биноан бош қисмга, яъни head тегида фақат веб-хужжатимизни акслантиришда керак бўладиган маълумотларни ёзамиш. Куйида келтириладиган тавсиф хабарлар фақат ва фақат бош қисмга ёзиш тавсия этилади. Акс ҳолда барча браузерларда бир хил намоён бўлмаслиги мумкин. Масалан, кодламани фақатгина хужжатнинг илк 512 байти ичида эълон қилиш шарт. Акс ҳолда браузер ўзининг одатий қиймати бўйича ёки очилаётган хужжатнинг операциян тизим аниқлаган кодламасида кўрсатилгандаи акслантиради.

HTML нинг 5-талқинида кодламани кўрсатиш синтаксиси анча соддалаштирилган. У қуйидагича ёзилади:

```
<meta charset="utf-8">
```

¹ Тавсиф хабар – уни мета-маълумотлар деб ҳам юритилади. Аммо биз хабар деганда катта ҳажмли маълумотни эмас, фақат асосий мазмунни ифодаловчи қисқа ахборотни тушунамиз. «Мета» деган олд кўшишчанинг аниқ таржимаси мавжуд эмас, *meta-data* – ХУЖЖАТ ДАСЛАВИДАИ ВА НИМАДАН ИБОРАТЛИГИНИ АНИҚЛОВЧИ ТАВСИФ-ХАБАР ДЕГАН МАЗМУНИ БИРЛАШДИ

Ташқи файлларнинг кўрсатишида тўлиқ URL ишлатилиши ёки жорий хужжатнинг жойлашган ўрнига нисбатан йўли кўрсатилиши мумкин. Масалан бирор расмни кўрсатиш учун, уни жорий файл ёнидаги «img» жилдига жойлаган бўлсақ, оддийгина «/img/rasm.png» кўринишида унга мурожаат қилинади. Бу албатта ёзилаётган кодни қисқа бўлишини таъминлаш билан бирга домен ўзгарганда файлни топа олмай қолиш хатолигидан ҳам сақлади. Агар веб-саҳифа яратиш жараёнида бир неча манбадан ишорат кўрсатиш зарурияти бўлса, унда ташқи жилдлар рўйхатини **base** тегида кўрсатилиши керак.

```
<base href=ишорат [target=номланиш]>
```

- **ишорат** – ташқи манбанинг каталог манзили.
- **номланиш** = _blank | _self | _parent | _top – очилаётган ишорат юкланиши керак бўлган ойна ёки фрейм номи.
 - _blank – ишоратни браузернинг янги ойнасида очади.
 - _self – (одатий қиймат) ишоратни браузернинг жорий ойнасида очади. Фрейм [5.7.] мавжуд бўлмаса, қўйидаги аломатлар ҳам шу сингари ишлайди.
 - _parent – фрейм ичидаги ишоратни ташқи саҳифада очади.
 - _top – фреймларни бекор қилиб, ишоратни жорий ойнада очади.

Ташқи файлларни боғлаш учун **link** тегидан фойдаланилади. Унинг тўла синтаксиси қўйидагича:

```
<link href=ишорат [type=файл тури][charset=кодлама]  
[rel=муносабат][title=матн][hreflang=тил белгиси]  
[sizes=ўлчам][media=қўрилма тури]>
```

- **ишорат** – боғланаётган файлнинг манзили.
- **файл тури** – файлнинг MIME (*Multipurpose Internet Mail Extension* – интернет почтасининг кўп мақсадли кенгайтмаси) тури: `text/html`, `text/css`, `image/gif` ва ҳоказо

күринишида бўлади. Кўрсатилган файл аслида нима эканлигини англатади. Номланиш бир қараганда тушунарли, лекин унинг тўлиқ ва тўғри рўйхатини аниқлаб бўлмайди, у доим кенгайиб бораверади. HTML5да agar `rel` кўрсатилган бўлса, файл турини кўрсатиш шарт эмас.

▪ **кодлама** – кодлама номи. Агар бошқасини ишлатишга жиддий сабаб бўлмаса, UTF-8 дан фойдаланиш тавсия этилади. Бу турли тиллардаги ҳарф ва белгиланишларни тўғри акслантиришга кафолат беради.

▪ **муносабат** – веб-саҳифага нисбатан унга боғланаётган файл қандай муносабатдалигини аниқлайди. Бунда ёзиладиган калит сўзлар HTMLда мавжуд тушунча бўлса, боғланаётган файлни ўша тушунчада ифодалангандай таҳлил қилиб юқланади. Масалан, унга «stylesheet» деб ёзилса, CSS жадвали сифатида қаралади; «alternate» деб ёзилса, у HTMLда эмас унга бошқа жойлаштирув тиллари(XML ёки бошқа)да ёзилган ҳужжат сифатида қаралади.

▪ **матн** – боғланган файл сарлавҳаси.
▪ **тил белгиси** – иккита ҳарфдан иборат тил белгиланиши, масалан: `uz`, `ru`, `en`.

▪ **ўлчам** – бу аломат HTML5 да киритилган, лекин браузерлар уни кўллашни бошлашгани йўқ. Одатда кўрсатилган расм қандай катталиқда бўлмасин уни браузер 16x16 ўлчамгача кичиклаштириб веб-саҳифа нишончаси сифатида акслантиради. Математик нуқтаи назардан умумжаҳон келишувига асосон биринчи эни, кейин лотинча «x» ҳарфи, сўнгра бўйи нуқталар ҳисобида ёзилади. Ҳозирча ушбу аломатдан фойдаланиш тавсия этилмайди.

▪ **курилма тури** – файл қандай курилмалар учун ишлashingини аниқлайди:

- `all` – барча курилмалар, бу унинг одатий қиймати.
- `braille` – кўзи ожизлар учун Брайл тизими асосида ишловчи курилма.
- `handheld` – экранни кичик смартфон ёки шунга ўхшаш курилмалар.
 - `print` – чоп этиш курилмалари.
 - `speech` – овозли курилмалар, матнни ўқувчи браузерлар

- screen – монитор экранни.
- projection – проекторлар
- tty – имконияти чекланган экранлар, терминаллар, ихчам қурилмалар
- tv – телевизорлар

LINK теги жорий веб-саҳифа аксланиши учун керак бўладиган қўшимча файлларни бириттириш учун ишлатилганилиги сабабли ҳам ва у HTMLнинг бош қисмида ёзилиши керак бўлган тег бўлгани учун ҳам унга ёзилган маълумотлар, веб саҳифага нишонча қўйишдан бошқа ҳолларда ҳеч бир кўринишида браузерда акслантирилмайди. Фақат янгиликларни қисқача даракларини кўрсатиб борувчи RSS (*Really Simple Syndication*) ёки бошқа шунга ўхшаш XML файллари кўрсатилган бўлса, фақат Opera браузери уни ишорат ёзиладиган ойначанинг ўнг қисмида title аломатида ёзилган ном бўйича кўрсатади. Қолган ҳолларда ёзилган маълумотлар фақат браузер ва HTMLга ишлов берувчи дастурлар учун муҳим аҳамият касб этади.

Бош қисмда ёзилиши керак бўлган meta тегида саҳифанинг қайси тилдаги, муаллифи ва бошқа хусусиятларини аниқлаш мумкин. Аммо HTML 5-талқинида кўп бундай хусусиятларни аниқлаш вазифаси янги киритилган тег ва аломатларга юклатилганилиги сабабли, унга кўп тўхталашибиз, фақат синтаксисини келтириб ўтамиш:

```
<meta [name=ном] [http-equiv=маркиб тури] [content=маркиб]>
```

Веб-саҳифа учун тилни кўрсатиш, излаш тизимлари ёки бошқа дастурлар лотин ҳарфларида ёзганимизда уни туркча, кирилл ҳарфларида ёзганимизда эса русча деб хато аниқлаб олмаслиги учун жуда муҳим. Тилни мета тегида эмас, ҳужжат турини аниқловчи «doctype»дан ташқари барча кодни ўз ичига олувчи html тегида аниқланади. У қуидаги аломатларга эга:

- manifest – саҳифа юкланаётганда бажарилиб турувчи дастур манзили.
- lang – тил белгиланиши.

Веб-саҳифанинг сарлавҳаси, аникроғи, номланиши

title теги ёрдамида берилади. Ушбу ном браузернинг энг юқори қисмида ношончадан сўнг кўринади.

Юқорида баён этилган билимларга асосланиб, бош қисмга каттароқ эътибор қаратган ҳолда энг кичик HTML кодини ёзамиз:

```
<!doctype html>
<html lang="uz">
    <head>
        <meta charset="utf-8">
        <base href="http://yurt.uz/" target="_self">
        <link rel="alternate" type="application/rss+xml"
              title="RSS дараклар" href="/rss">
        <link rel="alternate icon" href="ico.gif">
        <title>HTML 5 ва CSS 3 сабоқлари</title>
    </head>
    <body>
        HTML5дан илк сабоқ
    </body>
</html>
```

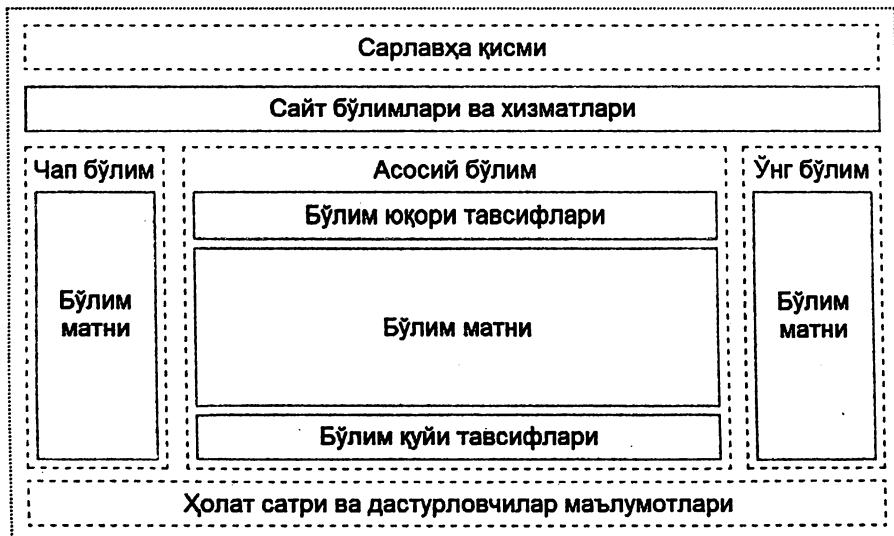
2.5. Сахифанинг асосий қисми

Асосий қисм, яъни **body** бўлимида ёзиладиган ҳар бир тег браузерда маълум обьект яратади. Аслида **body** тегининг ўзи ҳам браузернинг тўғри тўртбурчак шаклидаги намоиш майдонини тўлиқ эгаллайди. Аммо барча обьектлар ҳам кўринадиган бўлмаслиги ва ҳаттоқи алоҳида-алоҳида жойланган бир неча бошқа обьектларининг устидан ўрин олиши мумкин, яъни у устки қатламда акслантирилган бўлади. Нима бўлганда ҳам барчаси, **body** тегининг ички элементлари ҳисобланиб, **body** энг остки қатламдан ўрин олади. Унинг барча имкониятларини ўзига кўчириб олган **div** мукаммал тег ҳисобланиб, жойлаштириш тилида муҳим аҳамият касб этади.

Веб-сайт яратар эканмиз, аввало у қандай бўлимлардан ташкил топган бўлиши кераклигини аниқлаб, унга мос равишда экранни тўғри тўртбурчак шаклидаги бўлакларга бўлиб олиш лозим. Эскиз тайёр бўлгандан кейин, браузер ўзи ишлаётган компьютер ва монитор,

операцион тизим ва фойдаланувчиларнинг танловига қараб турли ўлчамда бўлиши мумкинлигини инобатга олган ҳолда, ҳар бир объектнинг ҳажмини ва барча ўлчамларнинг ўзгарувчанигини ҳисобга олиб жойлаштириш тилида код ёзишни бошлаган маъкул.

Сайтлар умумий ҳолда қуидагича қўринишда бўлади:



HTML нинг аввалги талқинларида ҳар бир жойлашувни аниқловчи тег бўлмаган, шунинг учун дастурловчилар **div** тегини турлича ҳусусиятлар билан ҳамма ерда қўллашга мажбур бўлишган.

5-талқинда жойлаштиришни қулайлаштирувчи ва ягона қолипга солувчи маҳсус теглар киритилган. Юқоридаги қўринишга эга веб-сайтимизни мавжуд теглар асосида чизиб чиқамиз. Қайси қисм керак бўлмаса, уни ҳосил қилувчи теглар ёзилмаслиги кифоя.

Сайтнинг бош сарлавҳаси, кичик мавзуси, соҳаси ва бошқа ҳусусиятларининг номланишлари одатда **h1**, **h2**, **h3**, **h4**, **h5** ва **h6** теглари ичida ёзилиши тавсия этилади. Уларнинг ягона аломати мавжуд:

```
<hN align = left | center | right | justify></hN>
```

- **left** – одатий қиймат, чап тарафга текислайди.
- **center** – матн майдон ўртасига жойлаштирилади.
- **right** – ўнг тарафга текислайди.
- **justify** – бир неча сатрли ёзувларни икки тарафга ҳам текислайди.

Энг катта сарлавҳа **h1**да энг кичиги **h6**да ёзилади. Уларни умумлаштириш учун **hgroup** теги ичида ёзиш энг мақбул йўл ҳисобланади.

Сайт сарлавҳаси, тамғаси, шиор ва тарғиботлари билан биргаликда унинг бўлимлари, излаш ва бошқа асосий хизматларини ўз ичига олган қисм, сайтнинг бош қисми ҳисобланади ва у **header** теги ичида ёзилади.

Асосий қисмда ишоратлар «а» теги ёрдамида берилади. [3.5.] Сайтнинг бўлимлари ва хизматларини кўрсатувчи ишоратларгина жойлашган қисми **nav** теги ичида ёзилиши тавсия этилади. Бошқа ишоратларни **nav** ичига олиб умумлаштириш таъқиқланади.

Ўнг ва чапда жойлашган бўлимларни **aside** теги ичида ёзилиши керак.

Веб саҳифанинг асосий, такрорланмайдиган, урғу бериладиган қисми **main** теги ичида ёзилиши лозим. Ушбу тег битта саҳифа учун ягона бўлиши ва мос ишорат бўйича ўзига хос маълумотни ичига олиши шарт.

Саҳифанинг асосий мақола чиқадиган қисми **article** тегида берилиши керак. Ҳолат сарти, дастурловчиларнинг муаллифлик ҳукуқи, сайт қачон ташкил топгани ва шунга ўхшашиб маълумотлар **footer** тегида жойланиши талаб этилади. Бошқа зарур бўладиган бўлакларнинг теглари **section** ёрдамида ажратилиши керак.

Юқоридаги келишувларни инобатта олган ҳолда саҳифанинг асосий қисмини қуийдагича таркибда ёзишмиз мумкин:

```
<body>
  <header>
    <hgroup>
      <h1 align="center">HTML 5 дан илк сабоқ</h1>
      <h4 align="right">Мукамал дастурлаш китобидан</h4>
      <h6 align="right">1-китоб</h6>
    </hgroup>
```

```

<nav>
    <a href="html.htm">HTML5</a>
    <a href="css.htm">CSS3</a>
    <a href="script.htm">JavaScript</a>
</nav>
</header>

<aside id="left">
    <p>HTML5</p>
    <p>CSS</p>
    <p>JavaScript</p>
</aside>

<aside id="right">
    <nav>
        <a href="abook.htm">Китоблар</a> <br>
        <a href="sbook.htm">Құлланмалар</a> <br>
        <a href="hbook.htm">Маълумотномалар</a>
    </nav>
</aside>

<main>
    <article>
        Уибы жойда саҳифанинг асосий
        маълумотлари жойлашади.
    </article>
</main>
<footer>
    © 2017
</footer>
</body>

```

Албатта у күтилғандай чиройли шаклда намоён бўлмайди. Уни ўзимиз хоҳлагандай чиқадиган қилишимиз учун CSSдан фойдалаган ҳолда маҳсус жилолар билан ёзиш керак*. Объектларга безак беришни кейинги мавзуларда батафсил кўриб ўтамиз.

26. Аломатларнинг қиймат турлари

Баъзи истесънолардан ташқари умумий ҳолда теглар күйидагида кўринишдаги синтаксисда ёзилишини лозимлиги билан танишдик:

<тег номи [аломат номи = қиймат] [...]> ... </тег номи>

Бунда «**қиймат**» битталик қўштироқ ичида берилиши ёки ажратувчи белгилар қатнашмаган ҳолда умуман қўштироқсиз ёзилиши мумкин, аммо турли чалкашликларга сабаб бўлмаслик учун, юқорида таъкидлаганимиздай иккиталик қўштироқ ичида ёзишни тавсия этамиз.

Баъзи тегларнинг қиймат бериш мажбурий бўлган аломатлари бўлади. Масалан, расмни кўрсатувчи тегга албатта расмнинг манзилини бериш шарт, акс ҳолда у тегнинг самараси бўлмайди. Яна бошқа тегларда эса аломатнинг номини ёзиш кифоя қиласди, уларга қиймат бериш шарт эмас. Ушбу ҳоллар фақат мантиқий «рост» ёки «ёлғон» қиймат қабул қиласдиган аломатларгагина тааллуқли. Бундай тег ва аломатлар билан келгусида батафсил танишамиз, ҳозир аломатларга бериладиган қийматлар қандай кўришинларда бўлишини аниқлаб олишимиз лозим. Бунинг учун қийматларни бир неча турларга ажратамиз:

▪ **Бутун (integer)** – бу элементар математикада қўлланиладиган бутун сон.

▪ **Ҳақиқий (real)** – бу фақат ўнли, яъни нуқта билан ёзиладиган математик ҳақиқий сон. Чексиз ва даврий касрлар ушбу турга кирмайди.

▪ **Сон (number)** – бу бутун ва ҳақиқий сонлар умумлашмаси, ўз ўрнида улар «–» ва «+» ишоралари билан ёзилиши мумкин.

▪ **Мантиқий тур (boolean)** – фақат «рост» ёки «ёлғон» қийматларидан бирини қабул қиласди. Рост қиймат: «true», «on», «yes» сингари сўзлардан бири орқали ифодаланади, ёлғон қиймат эса: «false», «off» ёки «no» сўзлари билан белгиланади. Агарда рост ҳам эмас, ёлғон ҳам эмас қиймат бериб, натижা қабул қилишни

браузернинг ўзига қўйиб беришга тўғри келса, бундай ҳолларда «auto» сўзини қўллаш лозим. Бунда натижа яна қандайдир қўшимча шартни бажарилишига қараб хисобланади.

▪ **Ўлчам** (size) – бу одатда «сон»дан кейин унинг қайси нисбий катталиқда эканлигини аниқлаш учун қўлланилади. Нол ҳар қандай катталиқда бирдай нол (0) ни англатгани учун ундан кейин «ўлчам» қўйиш шарт эмас. Нисбий катталиклар қуидагилардан иборат:

- **px** – пиксел, қурилманинг энг кичик ўлчамли элементи.

- **em** – мос шрифтдаги «m» ҳарфи эничалик катталиқ.

- **ex** – мос шрифтдаги «x» ҳарфи баландлигича катталиқ.

- **cm** – сантиметр.

- **mm** – миллиметр.

- **in** – дюйм, тахминан 2,54 сантиметр.

- **pt** – нуқта, бу ўлчам Macintosh тизимлари учун. Масалан 12 pt Macintoshда 12та нуқтага тенг бўлса, Windows тизимида 16 дан 22 гача оралиқда бўлиши мумкин.

- **pc** – пика, 12 та «нуқта» (pt, катталиги $\approx 35/83$ см)

▪ **Фоиз** – сондан кейин «%» белгисини қўйиш билан ифодаланилади. Унинг қиймати ташқи элемент ўлчамига нисбатан аниқланади.

▪ **Қисқа ўлчам** – ҳеч қандай катталиксиз фақат сон орқали ифодаланади. Агар соннинг ўзи берилса, ўлчамни шу сонча писел сифатида қабул қиласди. Агар сондан кейин «*» қўйилган бўлса, объектнинг ўлчами ёнидагилардан шунча маротаба фарқланишини англаради. Масалан, жадвалнинг битта устуни иккинчисидан уч марта катта бўлиши талаб этилса, биринчисига «1*» (ёки «*»нинг ўзи), иккинчисига «3*» қиймат берилиши карак.

▪ **Ишорат** – URL, унинг қандайлиги билан аввал танишганмиз. [2.3.]

▪ **Бурчак** – бурчак ва айлана ёйларини аниқлаш

учун сондан кейин қуидаги катталиклар ишлатилади:

- **deg** – градус. Геометрияда ўрганганимиздай, у 0 дан 360 гача қиймат қабул қиласы. Албатта ман-фий қиймат бериш ҳам мумкин, масалан: $-10\text{deg} = 350\text{deg}$

- **grad** – градиан. 0 – 400 оралиқдаги бутун сон-ларни қабул қиласы. 100 grad ≈ $\frac{1}{2}\pi$ түғри бурчакни англа-тади.

- **rad** – радиан. 0 – π оралиқдаги ҳақиқий сонлар-ни қабул қиласы. $1.5707\text{rad} \approx \frac{1}{2}\pi$ түғри бурчакни англа-тади.

- **turn** – қанча қисмга айлантиришни билдиради, 0 – 1 оралиқдаги ҳақиқий сонни қабул қиласы.

- **Вақт** – сондан кейин қуидаги катталиклар билан аниқланадиган ўлчов:

- **s** – сония.

- **ms** – миллисония.

- **Частота** – ўлчови сондан кейин қуидаги катталик-лар билан аниқланади:

- **hz** – герц.

- **khz** – килогерц.

- **Белги** – ASCII жадвалида келтирилган бир байт ҳажмли ягона код.

- **Сатр** – Күштирноқ ичида ёзиладиган бир қатордан иборат белгилар кетма-кетлиги. Сатр битталық ёки ик-киталик күштирноққа олинishi мумкин, қай бири билан очилса, ўша билан ёпилади. Шунинг учун ҳам бирининг ичида иккинчисидан фойдаланиш мум-кин, аммо ичма-ич құлланилмайди. Сатр ичида айни күштирноқ ишлатилиши керак бўлса «\» дан сўнг ёзи-лади ёки унинг маҳсус кодидан фойдаланилади. [2.1.]

- **Матн** – Хатбошилар билан бириктирилган «сатр»лар мажмуаси.

- **CDATA** – бу оддий (тег ва бўйруқларсиз) матннинг браузер акслантирадиган кўриниши демакдир. Унда сатрларни бўлувчи белгилар, табуляциялар умумлаш-тирилиб, битта бўш жой сифатида ёзилади. Маҳсус кодлар аксланадиган кўринишида келтирилади. Уни кўштирноқ ичида ёзилмайди, тегнинг якунини англа-

түвчи «</» белгилар жуфтлиги учраганда, CDATA ҳам ўзининг поёнига етганини тушунади.

▪ **Аниқланган тур** – Бундай тур бир қанча қабул қилинилиши мумкин бўлган «сатр»ларнинг тўпламидан ташкил топади. Нотўғри ёки аниқланмаган қиймат берилган ҳолда, браузер одатий қийматни олади. Масалан, матннинг уфқий жойлашишини аниқловчи align аломати фақат «left», «justify», «center» ва «right» қийматларидан биринигина қабул қила олади. Унга қиймат берилмаганда ёки ушбу тўртта сўздан бошқа нарса ёзилганда, одатий қиймати «left»ни қабул қилгандагидек ишлайди. [2.5.]

▪ **Ранг** – Тасвирланиш рангини белгиловчи катталик. У стандарт номи билан ёки RGB (қизил, яшил, кўк) тақсимоти ёрдамида, ўнолтилик саноқ тизими бўйича ҳар учала ранг учун нолдан FF гача қиймат берилиб, «#» белгисидан сўнг уларни бирлаштириб ёзиш орқали ифодаланади. Қизил (#FF0000) – red; Яшил (#00FF00) – green. (Муқованинг орқасида оммабоп ранг номлари билан RGB кодланишлари келтириб ўтилган)

▪ **MIME** – файл турини аниқловчи сатр. [2.4.]

▪ **Зарурмас қиймат**. Баъзи аломатларга қиймат бермаса ҳам унинг мавжудлигининг ўзи кутилган натижани беради. Масалан, танлашни англатувчи обьектдаги «checked» аломати, унинг тег ҳусусиятларида иштирок этишининг ўзи обьектнинг қиймати рост (белгиланган) эканлигини англатади. Агар бошқа аломатлар сингари бир хил андозада ёзиш лозим бўлса қиймат сифатида ўзининг номи берилади: checked="checked"

▪ **Идентификатор**. Ҳар бир тегнинг «id» аломатига саҳифа ичida такрорланмаслиги талаб этиладиган идентификатор бериш мумкин, фақат «name» аломатини номлашда эса «келишилган идентификатор»дан [2.1.] фойдаланса бўлади.

Сондан кейин кўрсатиласидиган катталиклар ҳеч қандай бўш жой ва ажратувчиларсиз, соннинг охирги рақамидан сўнг унга бириктириб ёзилади.

2.7. Универсал аломатлар

HTML тилида ҳамма теглар (одатда body тегида ёзилиши керак бўлган теглар назарда тутилади) кўллайдиган универсал аломатлар мавжуд. Уларни қўйида алифбо тартибида тавсифлари билан батафсил келтириб ўтамиз:

▪ **accesskey** – қиймати: белги, «0»дан «9»гача рақамлар ва «a»дан «z»гача лотин ҳарфларини қабул қиласди. Тугмалар бирикмаси босилганда браузердаги объект фоллашади ва унинг устида амал бажариш имконияти ҳосил бўлади. Аммо тугмалар бирикмаси ҳамма браузерда бир хил эмас. Ҳар бир браузернинг ўз фойдаланувчилари қандай тугмалар бирикмасини ишлатиш кераклигига кўникма ҳосил қилиб олган, лекин шу ўринда биз энг оммабоп браузерлар учун **accesskey="x"** қиймат берилганда, қандай тугмалар бирикмасини бошиш кераклигини келтириб ўтишни лозим топдик:

- Internet Explorer: Alt + X
- Microsoft Edge: Alt + X
- Chrome: Alt + X
- Opera: Shift + Esc, X
- Safari: Alt + X
- Firefox: Shift + Alt + X

Унутмаслик лозимки, ушбу брикмалар Windows операцион тизими учун берилган. MacOSда «Ctrl» тугмасини кўшиб босиш талаб этилади. Linuxда эса айрим бирикмалар ишламаслиги ҳам мумкин.

▪ **class** – қиймати: аниқланган турлар. CSS ёрдамида аниқланган жило номи берилади. Бир нечта номлар бўш жой билан ажратиб ёзилади. Номлар факат лотин ҳарфлари, рақамлар, чизиқча ва тагчизиқ (_) ёрдамида берилиши мумкин. CSS билан кейинги бўлимларда бағасил танишамиз*.

▪ **contenteditable** – қиймати: аниқланган мантиқий тур (true ёки false). Браузердаги объект қийматини фойдаланувчи ўзгартириш мумкинлигини аниқлайди. Аломатнинг ўзи ёзилиб, қиймати берилмаган ҳолда, унрист (true) деб қабул қиласди. Одатий қийматни эса ташки ҳечницидан ўзлаштириб олади.

▪ **contextmenu** – қиймати: аниқланган тур, унга menu теги ёрдамида яратилган рўйхатнинг IDсини берилади. Шунда объект устида ўнг тутмани босилганда чиқувчи менюга кўрсатилган рўйхат кўшилиши керак, афсуски HTMLнинг 5-талқинидаги ушбу ажойиб имконият ҳозирча ҳеч бир браузерда тадбиқ этилмаган.

▪ **dir** – қиймати: аниқланган тур. Матннинг намоён бўлиш йўналишини аниқлайди: ltr – чапдан ўнгта, rtl – ўнгдан чапга. Юникод кодламасида араб ва яхудийча тиллар йўналиши ўнгдан чапга деб бириктирилган, шунинг учун dir уларга таъсир қилмайди. Аммо бошқа ҳолларда матн йўналишини ўзгартириш учун «rtl»ни кўлласа бўлади, «ltr» эса унинг одатий қиймати.

▪ **hidden** – Объектни браузерда кўринмаслигини таъминлайди (зарурмас қиймат). Уни ишлатиш тавсия этилмайди, бундай имконият CSS ёрдамида берилгани маъқул, шунинг учун ҳам баъзи браузерлар уни кўлламайди.

▪ **id** – идентификатор. Объектнинг такрорланмас номини бериш учун фойдаланилади. [2.1.]

▪ **lang** – Объект ичida қайси тил белгиланишларидан фойдаланиш кераклигини англатади, қиймати: аниқланган тур. Шу ўринда стандарт бўйича қайси тиллардан фойдаланишини келтириб ўтамиз.

1-жадвал. Муомала тилларининг кодлари

Тил	код	Тил	код	Тил	код	Тил	код
Абхаз	ab	Дзонг-кэ	dz	Маорий	mi	Тожик	tg
Аймара	ay	Зулу	zu	Маратхи	mr	Тай (сиам)	th
Албан	sq	Иврит	he	Молдав	mo	Тамиль	ta
Амхар	am	Идиш	yi	Монгол	mn	Татар	tt
Англий	en	Индонезий	id	Науру	na	Тви	tw
Араб	ar	Интерлингва	ia	Немецкий	de	Телугу	te
Армян	hy	Интерлингве	ie	Непаль	ne	Тибет	bo
Ассам	as	Йоруба	yo	Норвеж	no	Тигринья	ti
Афан (оромо)	om	Ирланд	ga	Озарбайжон	az	Тонган	to
Афар	aa	Исланд	is	Окситан	oc	Тсонга(шангаан)	ts

Тил	код	Тил	код	Тил	код	Тил	код
Афгон (пушту)	ps	Испан	es	Ория	or	Турк	tr
Африкан	af	Итальян	it	Панджаби	pa	Туркмен	tk
Баск	eu	Каннада	kn	Персид (форс)	fa	Ўзбек	uz
Башкир	ba	Каталан	ca	Поль	pl	Уйғур	ug
Белорус	be	Кашмири	ks	Португаль	pt	Украин	uk
Бенгали	bn	Кечуа	qu	Ретороман	rm	Урду	ur
Бирман	my	Кинъяруанда	rw	Румын	ro	Фарер	fo
Бислама	bi	Корейс	ko	Рус	ru	Фиджий	fj
Бихари	bh	Корсикан	co	Саами	sz	Фин	fi
Болгар	bg	Коса	xh	Самоан	sm	Француз	fr
Бретон	br	Курд	ku	Санскрит	sa	Фриз	fy
Валлий	cy	Курунди	rn	Саньё	sg	Хауса	ha
Венгер	hu	Кхмер	km	Серб	Sr	Хинди	hi
Волапюк	vo	Қыргиз	ky	Сербохорват	sh	Хитой	zh
Волоф	wo	Қозоқ	kk	Сесото	st	Хорват	hr
Вьетнам	vi	Лаос	lo	Сетсвана	tn	Чеш	cs
Галисий	gl	Латвий	lv	Сингаль	si	Чивенда	ve
Голланд	nl	Латин	la	Синдхи	sd	Чишона	sn
Гренланд	kl	Лингала	ln	Сисвази	ss	Швед	sv
Грече	el	Литов	lt	Слован	sk	Эскимос (инт.)	iu
Грузин	ka	Лужицкий	sb	Словен	sl	Эскимос (юйт)	ik
Гуарани	gn	Македон	mk	Сомали	So	Эсперанто	eo
Гуджарати	gu	Малагасий	mg	Суахили	sw	Эстон	et
Гэль	gd	Малай	ms	Сундан	su	Яван	jv
Дат	da	Малаялям	ml	Тагаль	tl	Япон	ja
Джуанг	za	Мальтий	mt				

Баъзи тиллар бир нечта давлатларда турлича талқинда қўлланилади. Ҳатолик ва чалкашликлар юзага келмаслиги учун, ҳар бир тилни қўлланиладиган давлатига қараб белгилаш киритилган. Ушбу китоб мукаммал қўлланма бўлганлиги боис, ўқувчиликаримизга қўшимча адабиёт тавсия қилиши ўрнига тиллар жадвалини тўлалигича келтиришни маъкул топдик:

2-жадвал. Давлатлар бўйича муомала тилларининг кодлари.

Инглиз		Араб		Испан	
Австралия	en-au	Алжир	ar-dz	Аргентина	es-ar
Белиз	en-bz	Бахрейн	ar-bh	Боливия	es-bo
Великобритания	en-gb	Египет	ar-eg	Венесуэла	es-ve
Ирландия	en-ie	Иордания	ar-jo	Гватемала	es-gt
Канада	en-ca	Ирак	ar-iq	Гондурас	es-hn
Кариб ороли	en	Йемен	ar-ye	Доминикан	es-do
Янги Зеландия	en-nz	Катар	ar-qa	Испания	es
США	en-us	Кувейт	ar-kw	Колумбия	es-co
Тринидад,	en-tt	Ливан	ar-lb	Коста-Рика	es-cr
Тобаго	en-za	Ливия	ar-ly	Мексика	es-mx
ЮАР	en-jm	Марокко	ar-ma	Никарагуа	es-ri
Ямайка		ОАЭ	ar-ae	Панама	es-pa
Голланд		Оман	ar-om	Парагвай	es-py
Бельгия	nl-be	Саудия. А	ar-sa	Перу	es-pe
Нидерландия	nl	Сирия	ar-sy	Пуэрто-Рико	es-pr
Гэль		Тунис	ar-tn	Сальвадор	es-sv
Ирландия	gd-ie			Уругвай	es-uy
Шотландия	gd			Чили	es-cl
Италиян		Рус		Эквадор	es-ec
Италия	it	Молдавия	ru-mo		
Швейцария	it-ch	Россия	ru		
Хитой		Олмон		Француз	
Гонконг	zh-hk	Австрия	de-at	Бельгия	fr-be
КНР	zh-cn	Германия	de	Канада	fr-ca
Сингапур	zh-sg	Лихтенштейн	de-li	Люксембург	fr-lu
Тайвань	zh-tw	Люксембург	de-lu	Франция	fr
Швейцария	de-ch	Швейцария	de-ch	Швейцария	fr-ch
Португал		Румин		Швед	
Бразилия	pt-br	Молдавия	ro-mo	Финляндия	sv-fi
Португалия	pt	Руминия	ro	Швеция	sv

▪ **spellcheck** – қиймати: мантиқий тур (yes ёки no), браузер ёзув түғрилигини текшириш-текширмаслигиги аниқлайди. Одатий қийматини браузерда созлаш мумкин. Ушбу алматни ҳамма тег учун ёзиш мумкин, лекин самараси фақат матнини таҳирлаш мумкин бўлган объектлардагина билинади. Аломат эълон қилиниб қиймати берилмаган ҳолда, уни рост (yes) деб қабул қиласди. Афсуски, ўзбек тилини грамматикасини текшириш ҳали ҳеч қайси оммавий дастурларда кири-тилмаган.

▪ **style** – Объектга CSS жилоларини бериш учун фойдаланилади, қиймати CDATA.

▪ **tabindex** – Браузерда «Tab» тутмаси босилганда навбатма-навбат фаоллашиш кетма-кетлигини аниқлайди, қиймати мусбат бутун сон. «Табиндекс» кўрсатилмаган ёки «0» қиймат берилган объектлар яратилиш тартиби бўйича индексланган объектлардан кейин фаоллашади. Яширин ва нофаол қилинган объектлар бу кетма-кетликда иштирок этмайди. Индекси кичик объектдан индекси каттаси кейин фаоллашади, шунинг учун қатъий кетма-кет келадиган натурал сонлар ишилатиш шарт эмас, орада бир неча сон тушиб қолган бўлса ҳам, навбат бўйича кейингисига ўтиб кетаверади. Тескари тартибда кетма-кет танлаш учун одатда Shift+Tab биргаликда босилади.

▪ **title** – Объект устига сичқонча борганда чиқадиган изоҳчани аниқлайди, қиймати: сатр. Ўнинг кўринишини браузер ва ОТнинг ўзи аниқлайди, унга узундан-узоқ ва бир неча қаторли матнлар ёзиш тавсия этилмайди.

2.8. Ҳодисалар

Ҳодисалар – браузерда қандайдир амал бажарилаётган пайтда ё ўша лаҳзадан аввал ёки кейин бажарилувчи JavaScript кодларини аниқловчи бўйруқлар мажмуи. Ҳодисалар объектларга тегнинг алмати сифатида ёзилади, қиймати эса сатр қўринишида берилади. Ҳодисаларга бўйруқлар ёзишни JavaScriptга бағишиланган китобимиз-

да батафсил кўриб ўтамиз, ҳозирча фақат HTML аломатлари ҳисобланганлиги боис, ҳодисалар номларини санаб ўтишни маъқул топдик:

2.8.1. Саҳифага боғлиқ ҳодисалар (BODY тегида ёзилади):

- **onafterprint** – ҳужжат чоп этилгандан кейин
- **onbeforeprint** – ҳужжат чоп этилмасдан аввал
- **onbeforeunload** – ҳужжат юкламасдан олдин
- **onblur** – ҳужжат нофаол бўлганда
- **onerror** – ҳужжат юкланаётганда хатолик бўлса
- **onfocus** – ҳужжат фаоллашганда
- **onhaschange** – ҳужжат таркиби ўзгартирилганда
- **onload** – ҳужжат тўла юкланиб бўлганда
- **onmessage** – ҳужжат хабар қабул қилинганда
- **onoffline** – интернет билан алоқа узилиб қолганда
- **ononline** – ҳужжат интернетга уланганда
- **onpagehide** – ҳужжат кўринмас қилиб қўйилганда
- **onpageshow** – ҳужжат акслантирилганда
- **onpopstate** – ташрифлар тарихи ўзгартганда
- **onredo** – ташрифлар тарихидаги кейинги саҳифага ўтилганда
- **onresize** – ҳужжат ойнасининг ўлчами ўзгартирилса
- **onundo** – ташрифлар тарихидаги олдинги саҳифага ўтилганда
- **onunload** – ҳужжат ойнаси ёпилганда

2.8.2. Сичқонча ҳаракатига боғлиқ ҳодисалар:

- **onclick** – сичқонча босилганда
- **oncontextmenu** – матнбоп меню чақирилганда
- **ondblclick** – сичқончани икки марта босганда
- **ondrag** – сичқонча ёрдамида элемент ушланиб ҳаракатлантирилаётганда
- **ondragend** – сичқонча билан элемент ҳаракатлантирилиши тугаганда
- **ondragenter** – сичқонча билан объект устига элемент келтирилганда

- **ondragleave** – сичқонча билан ушланган элемент объект устидан кеттанды
- **ondragstart** – сичқонча билан элемент ушланганда
- **ondrop** – сичқонча билан объект устига элемент келтириб ташланганда
 - **onmousedown** – сичқонча босилғанда
 - **onmousemove** – сичқонча ҳаракатланаёттанды
 - **onmouseout** – сичқонча объект устидан кеттанды
 - **onmouseover** – сичқонча объект устига кирганды
 - **onmouseup** – босилған сичқонча қўйиб юборилғанда
 - **onmousewheel** – сичқонча ғиддираги айлантирилганда
- **onscroll** – объектнинг суриш тасмаси қўзғатилғанда

2.8.3. Клавиатурага боғлиқ ҳодисалар:

- **onkeydown** – тутма босилғанда
- **onkeypress** – тутма босиб турилған ҳолатда
- **onkeyup** – босилған тутма қўйиб юборилғанда

2.8.4. Форма ҳодисалари (FORM тегида ёзилади):

- **onblur** – форма нофаол бўлганда
- **onchange** – формада ўзгартириш бўлаёттанды
- **onfocus** – форма фаоллашганда
- **onformchange** – формада ўзгартириш бўлганда
- **onforminput** – фойдаланувчи формага маълумот киритаёттанды
 - **oninput** – фойдаланувчи форма элементига ўзгариш киритаёттанды
- **onselect** – форманинг элементи танланганда
- **onsubmit** – формадаги маълумотлар юборилғанда

2.8.5. Медиа ҳодисалар:

- **onabort** – файл юкланиши узилиб қолганда
- **oncanplay** – ижро этиш учун етарлича файлнинг қисми юкланиб бўлганда

- **oncanplaythrough** – охиригача узлуксиз ижро этиш учун етарлича юкланданда
- **ondurationchange** – файл узулиги ўзгарданда
- **onemptied** – файлни юклаш имконияти йўқолданда
- **onended** – файлни охиригача ижро этиб бўлганда
- **onerror** – файл юкланишда хатолик бўлса
- **onloadeddata** – файл юкланаётганда
- **onloadedmetadata** – тафсиф-хабарлар юкланаётганда
- **onloadstart** – файл юкланиши бошланганда
- **onpause** – ижро тўхтамга келтирилганда
- **onplay** – файл ижро этишга тайёр бўлганда
- **onplaying** – ижро этиш бошланганда
- **onprogress** – мета-хабарларни ҳисоблаётганда
- **onratechange** – ижро этиш тезлиги ўзгарданда
- **onreadystatechange** – файл ҳолати ўзгарганда
- **ontimeupdate** – вақт бўйича ўтказиб ижро этилганда
- **onvolumechange** – товуш баландлиги ўзгарганда
- **onwaiting** – ижро тўхтатилганда

III БҮЛІМ

МАТННИ ЖОЙЛАШТИРИШ



3.1. Хатбоши ва блоклар

Матнларни жойлаштиришда бир мұхим нарса-га әтебор қаратышимиз керак. Барча теглар, хусусан матнға шакл берувчи ва унинг қандай аксланишини аниқловчилар ҳам, икки тоифага бўлинади: блокли ва сатрли теглар [8.8.]. Блокли теглар* шуниси билан фарқланадики, улар ўзи жойлашган элемент майдонининг барча кенглигини эгаллайди, баландиги эса ичидаги обьектларга боғлиқ равишда аниқланади, шу билан бирга доим янги қатордан бошланади. Шунинг учун матннинг давомидан уланувчи сатрни блокли тегларга олинмайди.

Юқорида [2.5.] h1дан h4гача блокли тегларни кўриб чиқдик. Уларни матн орасида сарлавҳани ифодалашда қўлласа бўлади. Лекин ягона саҳифа бир неча мавзулардан иборат узундан-узоқ матндан иборат бўлиши мақсадга мувофиқ эмас. Ўқувчи битта чўзилган саҳифани айлантириб ўқиши жуда зерикарли. Битта мавзуу иккинчисидан яққол ажраб турмайди. Шунинг учун ҳар бир мавзуни алоҳида саҳифада очиладиган қилиш лозим. Буни инобатга олган ҳолда ушбу тегларни фақат саҳифанинг юқорисидаги бош қисмда қўллашни тавсия этамиз.

Юқорида br теги билан танишганимизда, уни кейинги қаторга туширишни таъминловчи тег сифатида баён этдик. Лекин бу дегани янги хат бошидан бошлашни англатмайди. Чунки одатда хатбошидаги биринчи сатр қолганларига нисбатан ичкарида жойлашади. ВR теги нафақат сатрларни, балки обьектларни ҳам пастки

қаторга туширишда қўлланилади. Сатрларнинг орасини очиш учун бир нечасини кетма-кет ёзиш ҳоллари ҳам кузатилади. Аслида бу нотўғри усул. Объетларни жойлашишини фақат CSS ёрдамида кўрсатилиши керак. Шундай экан ушбу BR тегини матнлардаги сатрларни бўлиш учунгина қўллаш тавсия этилади.

Агар хатбоши лозим бўлса
 сингари код ёзиш, аслида аксланиш бўйича талабга жавоб берса-да, мутлақо ноўрин. Хатбоши маҳсус тег p (параграф сўзидан олинган) блокли теги ёрдамида ифодаланади. У матнни қандай усулда текислашни кўрсатувчи аломатга эга: [2.5.]

```
<p [align="left | center | right | justify"]> ... </p>
```

Агар тегни ёпишни унугтан бўлсангиз, унинг таъсир доираси кейинги блокли тег очилган жойгача бўлади. Хатбошидаги биринчи сатр қолганларидан ичкарида бўлиши қандай таъминланишини CSS билан танишганда батафсил кўриб чиқамиз.

3.2. Матнни жилолаш

HTML тили матнни жойлаштиришга мўлжалланганлиги боис ҳам аксарият теглар матнга жило беришга йўналтирилган. Ҳар бир белги, ҳар бир сўз, ҳар бир қатор ва хатбошиларни; қолаверса бутун саҳифадаги матн қандай аксланиши лозимлигини сатрлар ичидаги теглар ёрдамида аниқлаб ёзиб борилади. (Ёзилмаган тақдирда одатий қийматлар асосида аксланишини айтиб ўтдик.) Матнни жилолаш учун хизмат қиласиган тегларнинг очилувчи ва ёпилувчи қисмлари ёзилиши қатъий талаб қилинади, таъсири ҳам унинг ички қисмида ёзилган матн учун амал қиласи.

Қуйида жило берувчи сатрли тегларнинг батафсил рўйхатини келтирамиз:

- **b** ва **strong** – қалин матн.
- **mark** – матннинг муҳим қисмини ажратиб кўрсатиш учун фойдаланилади.

- **big** – одатдагидан бир бирлик каттароқ ўлчамли матн.
- **cite, q** – кўчирмани ифодаловчи матн.
- **code, var** – дастурий кодларни англатади, барча ҳарфлар эни бир хил бўлади.
- **em, dfn** ва **i** – эгри матн.
- **ins** ва **u** – тагига чизилган матн.
- **kbd, tt** ва **samp** – энлик матн, ҳар бир ҳарфнинг эни бир хил ўлчамда бўлади.
- **small** – одатдагидан сал кичикроқ ўлчамли матн.
- **strike, del** ва **s** – (устидан) чизилган матн.
- **sub** – индексга туширилган матн.
- **sup** – даражага чиқарилган матн.

Юқоридаги рўйхатдан кўриниб турибдики, бაъзи бир неча теглар бир хил вазифани бажаарар экан. Бу турли хил матн редакторларида ишловчилар учун ҳам тушунарли бўлиш учун улардаги қисқартмалар билан олинган. Ушбу тегларнинг айримлари эскирган деб ҳисобланниб, қўллаш тавсия этилмайди. Аммо ҳеч биридан буткул воз кечилмагани учун уларнинг барчасини келтириб ўтишни лозим топдик. Амалиётда қўллашнинг ўрни келиб қолиши мумкин, аслида буларнинг ҳаммасини CSS ёрдамида ифодалаш мумкин. HTML тили илик яратилган вақтда CSS тушунчasi бўлмаган, шунинг учун ҳам матнларга жило бериш учун теглардан фойдаланилган. Матннинг рангини ва хуснихатини (шрифтини) ўзгартириш учун **font** теги қўлланилган. Унинг синтаксиси қўйидагича:

 матн

Бунда:

о ўлчам матннинг катталигини аниқлаб, фақат 1 дан 7 гача бутун сон қабул қиласди. Агар саҳифанинг бош қисмида **basefont** ёрдамида хуснихат ўлчами кўрсатилган бўлса, унга нисбатан FONTнинг ўлчамида, мос равишда «+1», «+2», ... деб матнни катталаштириш ёки минус билан кичиклаштириш мумкин. Албатта бу ерда қўшиш ёки айиришдан сўнг матн катталиғи 1 ва 7 нинг орасида бўлади деб қаралади.

Бошқача қиймат берилганда эса, HTML қоидаси бўйича одатий қийматда матн аксланаверади.

- ранг – [2.6.] бўлимда аниқланган ранглардан бири.
- ном ўрнига ОТда мавжуд ҳуснихатларнинг номи кўрсатилади. Мисол:

```
<font size="+1" color="blue" face="Tahoma">кўк матн</font>
```

Веб-саҳифаларда эълон қилинган маълумотлар қачон киритилганигини кўрсатиш учун, жорий вақтга нисбатан ёзув сифатида намоён этиш одат тусига кириб улгурди. Масалан Оврўпадаги сайтда чиққан мақола, уларнинг вақти билан кўрсатилгандан кўра, ўқувчининг браузери жорий маҳаллий вақтга нисбатан фалон соат ёки дақиқа аввал деб кўрсатиб турса аникроқ ва тушунарлироқ бўлади. Умуман олганда вақтни кўрсатиш учун **time** теги киритилган. Унинг «pubdate» деган аломати мавжуд бўлиб, маълумот киритилган вақт кўрсатилиши назарда тутилган эди. Лекин амалиёт натижасида ушбу аломат ортиқчалиги маълум бўлди ва уни қўллаш тавсия этилмайди. Вақтни кўрсатиш учун ягона «datetime» аломати кифоя этади. Maxsus скрипт орқали **time** теги ичидаги ёзувни жорий вақтга нисбатан хисоблаб, кўрсатиш мумкин. Мисол:

```
<time datetime="2017-04-29T18:24:11+05:00">5 дақиқа аввал</time>
```

Баъзи қадимий тилларда, хусусан араб ёзувида ҳам ўнгдан чапга қараб ёзилади, бу эса биз кўнинкан йўналишга тескари. Бундай матнларни акслантириш учун тескари тартибда ҳар бир қаторни аниқ ёзиб чиқиш жуда мураккаблик туғдиради. Бу муаммони бартараф этиш учун **bdo** теги киритилган:

```
<bdo dir="ltr | rtl"> ... </bdo>
```

- **ltr** – одатий қиймат, матн чапдан ўнгта қараб чоп этиб борилади.
- **rtl** – матн (right to left) ўнгдан чапга қараб чоп этиб борилади.

Агар йўналиши ўзгартирилган ёзув ичида бирор жумла асл ҳолида кўриниши шарт бўлса, уни **bd** теги ичида ёзиш лозим. Масалан, араб ёзуви ичида инглизча сўзни келтириш зарурияти туғилса, HTML5да қўшилган ушбу тег орқали ортиқча мураккабликсиз буни амалга ошириш мумкин. (Афсуски Internet Explorer ва Edgeларнинг ҳозирги талқинлари уни қўлламайди.)

Умуман олганда, CSS ёрдамида жило бериб **span** теги ичида сатрни исталганча кўринишда акслантириш мумкин.

3.3. Эскирган теглар

1996 йилдан HTMLда кодларни қисқартириш, соддатушунарлироқ ёзув услубига ўтиш учун ҳам CSS киритилиб, кенг кўлламда қўлланила бошлади. Уни такомилаштириш билан матнларни жойлаштириш ва уларга жило беришга мўлжалланган кўпгина HTML тегларни қўллашта эҳтиёж қолмади. Бундан ташқари баъзи тегларни ўрнини босадиган, оммабоп тарзда номланган ва ягона стандартга келтирилган янги теглар қўшилди. Ушбу тараққиёт босқичлари баъзи тегларни эскирган деб қарашни тақазо қиласди. Улардан фойдаланиш тавсия этилмайди, аммо олдин ёзилган кодлар ўз вазифасини бажариши учун ҳам ҳамма брузерлар уларни қўллайди. Шундай бўлса ҳам, яқин келажакда улардан бутунлай воз кечилиши мумкин, ёки улар ягона стандартга келтирилмаганилиги учун ҳам турли браузерларда турлича аксланиши мумкин. Агар янги DOCTYPEда инобатга олинмаган бўлса, баъзи қурилмаларда хато код сифатида эътиборсиз қолдирилиши ҳам мумкин. Шунинг учун W3C тавсияларини инобатта олишимиз мухим.

Куйида эскирган (HTML5да қўллаш умуман тавсия этилмайдиган) тегларни кўриб чиқамиз, улардан фойдаланмаслигимизни инобатта олган ҳолда тўлиқ синтаксиси келтирилмайди:

- **acronym** – қисқартмалар (HTML, CSS, JS ва бошқалар). Унинг ўрнига **abbr** фойдаланилади.

▪ **applet** – Java дастурлаш тилида ёзиладиган кодлар. Ушбу тегнинг очилувчи ва ёпилувчи қисмлари орасига, яъни тег асосига матн ёзилса, у «applet»ни кўлламайдиган браузерларда аксланади. Кўлловчи браузерлар коднинг натижасини кўрсатади. Унинг ўрнига универсал хисобланган **embed** тегидан фойдаланиш тавсия этилади.

▪ **center** – ичидаги элементларни ташқи объектга нисбатан уфқий марказда жойлаштиради. Уни барча браузерлар қўллагани учун қўпгина дизайнерлар ҳали ҳам ундан кенг фойдаланишади. Аслида ўрнига CSS қоидасини киритиш лозим.*

▪ **dir** – ОТ файл ва жиллар рўйхатини кўрсатиш учун мўлжалланган. Унинг ўрнига умумийлашган, рўйхат кўрсатувчи **UL** теги қўлланилади.

▪ **frame** – бошқа HTML файлни жорий саҳифада кўрсатиш. Бундай йўл аслида нотўғри, лекин баъзан саҳифанинг бир қисмида бутунлай бошқа жило ва қонуниятлар билан ёзилган манъбадаги маълумотларни кўрсатишга тўғри келиб қолиши мумкин. Агар ундағи кодларини жорий саҳифага, айтайлик оддий **DIV** теги ичига жойлаштирилса, табиийки ўз жозибасини йўқотади ва кўриниши бузилиб кетади. Шунга ўхшаш ноchor ҳолларда фреймдан фойдаланишга мажбур бўлиниади. Лекин **frame** ўрнига **iframe** қўллаш мумкин.

▪ **frameset** – бир қанча фреймлардан ташкил топган саҳифаларда қўлланилган. Агар у ишлатилса **BODY** тегидан фойдаланмасликка тўғри келади.

▪ **isindex** – маълумот киритиладиган ойна олдиға ёзилаган жумлани битта сатрда бўлишини таъминловчи, индексланган матн. Уни саҳифанинг бош (**HEAD**) қисмида эълон қилиб қўйилади. Кейин браузер ушбу жумлани қаерда учратса, битта сатрда, тепа ва пастки қисмига чизик чизиб акслантириши керак. Яъни битта жумла икки марта ёзилиши тақазо қилинади, бу эса мукаммал дастурлаш талабларига мос келмайди. Маълумот киритиш ойнаси қўлланилиши лозим бўлса, **input** ва **label** тегларидан фойдаланиш кифоя. **ISINDEX** тавсия этилмайдиган

тег бўлганлиги боис браузерлар кутилгандагидек на-
мойиш этмайди.

▪ **menu** – бу ҳам рўйхатни акслантиришга мўлжаллан-
ган ва унинг ўрнига ULдан фойдаланилади. HTML5да
menu теги бутунлай бошқа мақсадда киритилган.*

▪ **nextid** – браузер ва HTML таҳрирловчилари учун ке-
йинги саҳифани топиб олиш учун кўлланилган. Ҳозирда
умуман ишлатилмайди.

▪ **nobr** – ичига ёзилган матн ва элементларни фақат
битта қаторда чиқишини таъминлайди. У маъно
жиҳатдан ётга тескари вазифани бажаради. Ўрнига CSS
кўллаш тавсия этилади.

▪ **noembed** – агар браузер тегишли плагинни (бра-
узернинг имкониятларини оширувчи қўшимча дастур,
масалан: Flash, QuickTime ва бошқаларни) кўлламаса,
тег асосидаги тасвир кўринади, акс ҳолда тег эътиборсиз
қолдирилади.

▪ **noframes** – ичида ёзилганлар, фреймларни
кўлламайдиган браузерларда аксланади. Барча замона-
вий браузерлар фреймни кўллагани боис уни эътибор-
сиз қолдиради. Шунинг учун ҳам уни ишлатишнинг
маъноси йўқ.

▪ **object** – браузер тушунмайдиган кодлар (товушлар,
лавҳалар, дастурлаш кодлари ва бошқалар) қандай акс-
лантирилиши лозимлигини аниқлайди. Унинг ўрнига
универсал **embed** теги ишлатилади.

▪ **plaintext** – матн қандай ёзилса шундай, HTMLнинг
шартли ўзгартиришларисиз, бўш жой ва бўш сатрларни
ўз ўрнидалигича акслантириши. Унинг ўрнига PRE теги
ёки алоҳида файл сифатида юкланса, «text/plain» МИМЕ-
тури кўлланилади.

▪ **xmp** – код қандай ёзилган бўлса, шундайлигича
акслантиради. Унинг PREдан фарқи шундаки, ёпил-
магунча ичидағи тегларни ҳам ёзув сифатида кўрса-
таверади. Эҳтимол шунинг учун ҳам буни ишлатиш
тавсия этилмас, аммо HTML ва дастурний кодларни
чиқариш учун қулай восита. Бошқа ҳолларда «<» ва
«>»ларни «<» ва «>»лар ёрдамида ёзив чиқишига
тўғри келади.

- Күйидаги теглар бевосита CSS ёрдамида ифодаланади:
- **big** – хуснихатни бир бирликка катталаштиради.
 - **listing** – кодларни чоп этиш учун мұлжалланган, хуснихатни кенгайтирилған ва сал кичиклаштирилған ҳолда күрсатади.
 - **font** – хуснихатни ўзgartириш. [3.2.]
 - **rb** – ruby тегидаги асосий матнни ажратиш учун фойдаланилған. Ҳозирда умуман қўлланилмайди.
 - **blink** – пирпирайдиган (ёниб-ўчадиган) матн.
 - **basefont** – сахифанинг бош (HEAD) қисмида FONTдан ташқари барча теглар учун қўлланиладиган хуснихат номи, ўлчами ва рангини аниқлайди.
 - **strike, s** – устига чизилған матн.
 - **tt** – кенгли ёзув. Ҳар бир ҳарф ёки белгининг эни бир хил ўлчамда бўлади. Унинг ўрнига «Consolas» ёки «Courier New» каби хуснихатларни қўллаш мумкин.
 - **u** – тагига чизилған матн.

Булардан ташқари эскирган аломатлар ҳам мавжуд. Уларнинг барчаси CSS ёрдамида ифодаланади. Масалан BODY тегида сахифанинг қандай шаклланишини аниқловчи background, bgcolor, link, alink ва vlink сингари аломатларни қўллаш тавсия этилмайди. Ҳар бир тег учун қайси аломати эскирган эканлигини тўлиқ рўйхат сифатида келтириб ўтмаймиз. Тегнинг синтаксисини тўла баён этаётганимизда уларни изоҳ сифатида келтирамиз ёки умуман тушириб қолдирамиз. Фойдаланилмайдиган билим учун батафсил тўхталишга ҳожат йўқ.

3.4. Рўйхат

Сахифада баъзан нимадирларни рўйхатини санаб кўрсатишга тўғри келиб қолади. Ўмуман олганда кетма-кет ёзиладиган сатрларни рўйхат сифатида ифодалаш кўп ҳолларда қулийлик туғдиради. HTMLда икки хил кўринишда рўйхат мавжуд: саноқли рўйхат ва белгиланган рўйхат.

3.4.1. Саноқли рўйхат

Бунда келтириладиган элементлар ҳар бири янги сатрда тартиб рақам билан акслантирилади. Синтаксиси қўйидагича:

```
<ol [type="A | a | I | i | 1"] [reversed] [start="бутун сон"]>
    <li> сант 1 </li>
    <li> сант 2 </li>
    .
    .
    .
</ol>
```

Бунда type рўйхат қандай тартиб рақамига эга бўлишини аниқлайди:

- A – Лотин алифбосининг бош ҳарфлари билан;
- a – Лотин алифбосининг кичик ҳарфлари билан;
- I – Рим рақамлари ёрдамида;
- i – Рим рақамлари билан фақат кичик ҳарфлар ёрдамида;
- 1 – Одатий қиймат, араб рақамлари билан ўнлик саноқ тизимида.
- start – аломатига тартиблашнинг бошланғич қийматини бериш мумкин.
- reversed – аломати қийматта эга эмас, агар у кўрсатилса, рақамлашни тескаридан камайиш тартибида акслантиради. Масалан 1, 2, 3 ўrniga 3, 2, 1.

HTML	<pre><ol type="A" reversed start="C"> саноқли рўйхат белгиланган рўйхат </pre>	Натижা	D. саноқли рўйхат С. белгиланган рўйхат
------	---	--------	--

3.4.2. Белгиланган рўйхат

Аммо ҳар доим ҳам тартиб рақамини кўрсатиш зарур бўлавермайди. Санаб ўтилаётган элементларнинг барчаси бир хилда аҳамиятга эга бўлса уларни ҳатбошидан ажратиб кўрсатиш керак бўлган ҳолларда белгиланган рўйхатдан фойдаланилади. Унинг синтаксиси қўйидагича:

```

<ul [type="disc | circle | square"]>
    <li> camp 1 </li>
    <li> camp 2 </li>
    . . .
</ul>

```

Бунда **type** рўйхат қандай белги билан кўрсатилишини аниқлайди:

disc — доира; **circle** — айлана; **square** — квадрат майдонча.

HTML	<pre> <ul type="square"> саноқли рўйхат белгиланган рўйхат </pre>	Натижা
		<ul style="list-style-type: none"> ▪ саноқли рўйхат ▪ белгиланган рўйхат

HTML5да рўйхатни **menu** теги орқали ҳам ифодалаш мумкин, лекин уни шунчаки рўйхат эмас, ҳақиқий меню сифатида фойдаланиш учун киритилган. Яъни, тутмани босганда чиқадиган кетма-кет ёки ёнма-ён турувчи (одатда ойналар тепасига қўйиладиган) тутмаларни намоён этишга мўлжалланган. Оддий ҳолларда «**ul**»нинг ўрнига «**menu**» ёзиш кифоя, фақат унинг «**type**» аломати бошқача қийматлар қабул қиласди:

- **context** — кетма-кет чиқувчи меню (сичқонча ўнг тутмаси босилгани сингари)
- **toolbar** — ёнма-ён чиқувчи меню (бошқарув панели сингари)
- **list** — шунчаки рўйхат, одатий қиймат, «**ul**»дан фарқи йўқ.

3.4.3. Изоҳли рўйхат.

Изоҳли рўйхат асосан атамалар ва уларни тавсифлаш учун қўлланилади. Рўйхат **dl** теги ёрдамида яратилади. Номланиш **dt** теги билан берилади, унга изоҳ эса **dd** ёрдамида ифодаланади.

Мисол:

```
<dl>
  <dt>Firefox</dt>
  <dd>Энг қулай браузер</dd>
  <dt>Chrome</dt>
  <dd>Энг тезкор браузер</dd>
</dl>
```

Ушбу учала тег алоҳида аломатларга эга эмас, уларни кўринишини CSS ёрдамида ўзgartириш мумкин.

3.4.4. Изоҳи яширин рўйхат

Кўп ҳолларда сарлавҳаларнинг ўзини рўйхат сифатида келтириш кифоя бўлади, аммо исталган пайтда мавзу нима ҳақидалигини билиб олиш учун яширин изоҳни очиб кўриш имкониятини ҳам қолдиришта тўғри келади. Бундай ҳолларда HTML5да янги киритилган, изоҳи яширин мавзуларни ёзишга мўлжалланган **details** тегидан фойдаланиш мақсадга мувофиқ. Унинг ичida қўлланиладиган **summary** тегида сарлавҳа ёзилади, қолган барча ёзув яширин тарзда бўлади. Фойдаланувчи истаган пайтда «батафсил» деган маънони билдирувчи милни босиб, яширин изоҳни кўриш мумкин. сарлавҳа билан изоҳи кўриниб туриши учун **details** тегида зарур мас қийматли «open» аломатини кўрсатиш керак.

Мисол:

```
<details>
  <summary><b>HTML</b></summary>
  <i>Жойлаштириши тили</i>
</details>
<details open>
  <summary><b>JavaScript</b></summary>
  <i>Дастурлаш тили</i>
</details>
```

- ▶ HTML
 - ▼ JavaScript
- Дастурлаш тили

3.5. Ишорат

Ишорат HTML тилида алоҳида ўринга эга муҳим тег ҳисобланади. Шунинг учун ҳам уни алифбонинг энг биринчи ҳарфи билан номланган. Аслида ишорат бошқа саҳифани юклаш учун хизмат қиласди, лекин замонавий талаблар айни бир саҳифанинг биргина бўлаги ўзгариши учун уни қайтадан юклаш самарасиз эканлигини кўрсатди. Шунинг учун эндиғи саҳифаларда талабга қараб, ишорат умуман учрамаслиги ҳам мумкин. JavaScript ёрдамида саҳифани тўлиқ қайта юкламай, серверга сўров юбориб, айнан қайтган маълумотларни акслантириш амалиёти мавжуд. Бу хусусида кейинги китобда батафсил ёритилади.

Ишорат барибир ўз долзарблигини йўқотмаган, браузерлар уларни маҳсус кўринишда акслантиради, фойдаланувчига кўринарли бўлишини тамиnlайди. Айни бир саҳифанинг манзилини статик равищда кўрсатиб қўйиш — уни тезда топиб очишида самара беради. Шуларни инобатга олган ҳолда, ишоратдан воз кечиб бўлмайди.

Ишорат қўйиш синтаксиси:

```
<a name="ном" href="ишорат" [аломатлар]>матн</a>
```

Ушбу тег асосан икки мақсадда фойдаланилади:

1. Кўрсатилган ишоратдаги саҳифани очиши учун.
2. Саҳифанинг шу тег ёзилган жойига лангар қўйиш учун.

Биринчи ҳолат энг кўп қўлланилади, унда матнга босилганда ишоратдаги саҳифа юкланди. Иккинчи ҳолатда эса саҳифанинг ўша жойига «лангар ташланади». Агарда саҳифа узун бўлиб, битта экранга сигмаётган бўлса, айтайлик иккинчи бобни ўқиш учун бир қанча айлантиришлар билан пастга тушишга тўғри келса, бундай ноқулайлик туғилмаслиги учун «лангар»дан фойдаланилади.

```
<a name="боб2">Иккинчи боб</a>
```

Юқоридаги сингари ёзилса, браузердаги манзил кири-

тиш дарчасига **http://...#bob2** деб мурожаат қилинганда, экранда пайдо бўлган саҳифа тепага сурилиб, айнан шу «Иккинчи боб» кўринадиган ҳолга келади. Агар белгиланган лангардан кейинги намоён бўладиган тасвирлар битта экран майдонидан кичкина бўлмаса ушбу ёзув экранда акслантирилаётган биринчи сатрда кўринади, акс ҳолда айлантиргич (scroll) энг пастта туширилади. Одатда саҳифа очилганда бошида мундарижа сифатида мавзулар келтирилади, уларда ишоратлар қуидагида ёзилади:

```
<a href="#bob2"> Иккинчи боб </a>
```

Шу мавзуга босилганда, ушбу саҳифани тепага сурилиб, қўйирогида жойлашган юқорида ёзилгандагидек лангар қўйилган қисми экранга кўринарли тарзда намоён бўлади. Таъкидлаб ўтишимиз жоизки, ушбу имкониятидан фойдаланишдан ҳали бутунлай воз кечилмаган бўлса ҳам у HTML5да эскирган хисобланади. Сабаби замонавий веб-саҳифаларда «#»дан кейин умуман лангарни англатмайдиган муҳим қийматлар бериш одатга кирган. Ҳозирда уни «хештег» деб аташ кенг оммалашиб кетди.

Ушбу тег **href** ва **name**дан ташқари яна бир қанча аломатларга эга:

- **charset** – ишоратдаги матн қайси кодламадалигини аниқлайди.
- **href** – ўтиш керак бўлган хужжатга ишорат, URL ёзилади
- **hreflang** – ишоратдаги матн тилини аниқлайди.
- **media** – кўрсатилган хужжат қандай қурилма учун мўлжалланганилиги [2.4].
- **name** – номи, ушбу ном билан браузер унинг жойлашган ўрнини топади.
- **rel** – жорий хужжатга нисбатан ишоратда кўрсатилган файл муносабати.
- **rev** – ишоратдагига нисбатан жорий хужжат муносабати (**rel**нинг акси).
- **tabindex** – «Tab» тугмаси босилгандаги фаоллашиш тартибини аниқлайди.
- **type** – ишоратдаги хужжатнинг МИМЕ тури. [2.4.]

Ушбу аломат фақат «`href`» билан бирга қўлланилади. Тур қўрсатилмаган ҳолда одатий қийматини ишоратдаги файл кенгайтмасидан ёки унинг бош қисмидаги тавсиф хабарда қўрсатилган турдан олади.

▪ **coords** – Фаол майдоннинг координатасини аниқлайди.

Ушбу аломатни ишлатиш учун аввал **map** теги ёрдамида саҳифадан майдон ажратиб олиш лозим.

▪ **shape** – фаол майдон шаклини аниқлайди. Талаб бўйича бу аломат билан `coords` бирга қўлланилади. Булар одатда саҳифага расм (ёки обьект) қўйиб, унинг қайсицир соҳасидагина ишорат фаол бўлишини аниқлашда фойдаланилади. Бунда қўрсатилган шакл билан бериладиган координата мутаносиб бўлиши керак. [5.1.]

▪ **download** – ишоратда қўрсатилган файлни юклаб олиш шартлигини англаради. Одатда браузерлар ўзида мавжуд ва қўшимча ўрнатилган дастурий таъминотлар асосида медиа файлларни ўзида қўрсатишни маъқул топади, бу эса ҳар доим ҳам мақул эмас.

▪ **hreflang** – ишоратда қўрсатилган ҳужжатнинг қайси тилдалигини қўрсатади.

▪ **target** – юкланаётган саҳифа қаерда аксланишини аниқлайди. [2.4.]

IV БҮЛІМ

ЖАДВАЛЛАР



4.1. Жадвални эълон қилиш

Жадвал — ҳисоботларни намоён қилиш учун әнг кенг күлланиладиган обьект. Аксланиши лозим бўлган матн ёки тасвиirlарни маҳсус сатр ва устунлар ёрдамида жойлаштириш учун жуда қулай восита. Агар жадвалнинг жойлашиши ва ўлчамлари учун аниқ, қатъий чегара қўйилмаса, у ичидағи маълумотлар ҳажмига қараб ўз кўринишини ўзгартириш ҳусусиятига эга. Баъзан бутун бошли веб-саҳифани бир жадвал ичига жойланганига ҳам гувоҳи бўламиз (2.5. даги андоза кўзда тутиляпти). Бундай усуслни қўллаш умуман тавсия этилмайди.

Жадвал `table` номли тег ёрдамида эълон қилинади, унда универсал аломатлар [2.7.] билан бирга фақат ўзига хос қўйидаги маҳсус аломатларини қўллаш ёки ўрнига CSSдан фойдаланиш мумкин:

- `align` — Уфқий жойлашишини аниқлайди. Аниқланган тур, қийматлари: `left`, `center` ёки `right`. Агар «`center`» қиймати берилса, жадвалнинг эни қандай катталиқда бўлмасин, икки томонидан четтача тенг ма-софа қолади. Бошқа ҳеч бир обьект бундай динамик ҳусусиятни ўзи намоён қила олмайди (эскирган `center` теги мундан мустасно), бунинг учун маҳсус CSSлардан фойдаланишга тўғри келади.*

- `background` — жадвалнинг орқасига қўйиладиган расм. Қиймати: ишорат.

- `bgcolor` — жадвалнинг ортранги. Қиймати: ранг.

- `border` — чегаравий чизиқлар қалинлиги. Қиймати: сон.

- **bordercolor** — чегаравий чизиқлар ранги. Қиймати: ранг.
- **cellpadding** — чегаравий чизиқдан катақдаги объектгача масофа. Қиймати: сон.
- **cellspacing** — катақлар орасидаги масофа. Қиймати: сон.
- **cols** — устунлар сони. Кўллаш шарт эмас, ўзи аниқлайди. Қиймати: сон.
- **frame** — жадвал атрофидағи ҳошия қандай кўринишда бўлишини кўрсатади. Қиймати – аниқланган тур:
 - **void** – жадвал атрофига чегара чизилмайди.
 - **border** – жадвал атрофига чегара чизилади. Одатий қиймат.
 - **above** – жадвалнинг юқори қисмига чегара чизилади.
 - **below** – жадвалнинг қуий қисмига чегара чизилади.
 - **hsides** – жадвалнинг қуий ва юқори қисмига чегара чизилади.
 - **vsides** – жадвалнинг ён тарафларига тик чегаралар чизилади.
 - **rhs** – жадвалнинг фақат ўнг тарафига чегара чизилади.
 - **lhs** – жадвалнинг фақат чап тарафига чегара чизилади.
- **height** – баландлиги. Қиймати: сон, фоизда ёки пикседа кўрсатилади.
- **rules** – катақларнинг қайси томонларида чегара чизилишини аниқлайди. Қиймати аниқланган тур:
 - **all** – катақларнинг барча чегаралари кўринади, **border > 0** бўлса одатий қиймат.
 - **groups** – фақат умумлаштирилган катақлар гурухларининг чегараси чизилади.
 - **cols** – устунлар орасидаги чегаравий тик чизиқларгина кўринади.
 - **none** – ушбу қиймат **border = 0** бўлганда одатий ҳисобланади.
 - **rows** – сатрлар орасидаги чегаравий чизиқларгина кўринади.
- **summary** – жадвалнинг қисқа тавсифи, бу браузерда аксланмайди, аммо изловчи дастурий таъминотлар бундан кенг фойдаланади. Қиймати: сатр.
- **width** – жадвал эни, қиймати сон, фоиз ёки пик-

сел ҳисобида. Агар жадвал катакларидағи объектларнинг умумий ҳажми кўрсатилган қийматдан ошиб кетса, ушбу аломат эътиборга олинмайди ва браузер янги қийматни ўзи танлайди.

4.2. Жадвалнинг сарлавҳаси

Жадвал **table** теги асосида яратилиб, фақат унинг ичидағина қўлланиладиган бир неча теглар ёрдамида қўшимча имкониятлари аниқланади. Жадвалнинг сарлавҳаси **caption** теги ёрдамида ёзилади. Аслида ушбу сарлавҳа **table** тегидан оддин ёзилган сатр кўринишида намоён бўлса ҳам, уни алоҳида ёзиш тавсия этилмайди, чунки аксланиш жараёнида сарлавҳа сифатидаги ёзув кутилгандагидан бошқа жойда пайдо бўлиши мумкин.

Аслида жадвал ичидағи барча объектлар аксланиб бўлгандан кейин, уларнинг ҳажмига қараб жадвал ўз ўлчамини ўзгаририб, браузерга мослайди. Баъзан устунлар хусусиятини ўзимиз аниқ киритиб қўйишимиз лозим бўлади. Бунда жадвал тўлик юкланишини кутмай, устунлар кўрсатилган шаклда намоён бўла бошлайди. Бунинг учун ҳар бир устунга мос равишда **col** теги ёрдамида омиллар белгилаб берилиши керак. **COL** теги аниқланган қиймат қабул қиласди:

- **align** – уфқий жойлашишини аниқлайди.

- **char** – қиймати: белги, устундаги барча қийматлардагиайнан шу белги битта тик чизиққа тўғриланиб ёзилади. Масалан, ҳақиқий сонлардан иборат устун бўлса, **char=''**, сингари ёзилганда устундаги барча сонлар шундай жойлашадики, улардаги ҳамма «» бир тик чизиқда ётади.Faқат юқоридаги align аломатига одатдагидан ташқари **char** қиймат берилгандагига (**align='char'**) бу ҳолат инобатга олинади. Афсуски ҳозирда ушбу ажойиб имкониятни ҳеч қайси браузер кутилгандагидек намоён этмайди. Келажакдаги браузерларнинг янги талқинларида буни кутиб қоламиз.

- **charoff** – устундаги қийматларни қанча белги кенглигига суришни аниқлайди. Қиймати бутун сон, ман-

фий бўлиши ҳам мумкин (бунинг тақдири ҳам `char` сингари).

▪ `span` – кўрсатилган аломатлар нечта устунга тадбиқ қилинишини аниқлайди.

▪ `valign` – тик кўринишда текислаш, қиймати аниқланган тур:

- `baseline` – қатордаги уфқий чизиққа текислайди (браузерлар қўлламайди).

- `bottom` – катакнинг қуи қисмига текислайди.

- `middle` – катакнинг ўрта қисмига текислайди, одатий қиймат

- `top` – катакнинг тепа қисмига текислайди.

▪ `width` – устун эни, қиймати сон, фоиз ёки пиксел ҳисобида.

```
<table cellpadding="5" cellspacing="0">
<col width="200" align="center" valign="top" span="3">
```

Айнан `col` билан деярли бир хил вазифани бажарувчи `colgroup` теги ҳам мавжуд. Уларнинг ягона фарқи шундаки, жадвалга `rules="groups"` аломати берилганда `col` билан гурухланган устунилар чегаралари чизилмайди, `colgroup` билан гурухланган устунилар чегаралари кўринади.

Бир неча қаторни сарлавҳа сифатида ажратиб олиш учун `thead` тегидан фойдаланилади. Асосан устуниларнинг номлари ёзиладиган сатрларни унинг ичида қўлланилади. Жадвалнинг охирги сатрлари, одатда жамловчи қийматлар ёзиладиган қаторлар `tfoot` теги ичида ёзилади. Қолган жадвалнинг асосий сатрлари `tbody` тегига олинади. Бу учала тег жадвални уч қисмга ажратиб, уларга алоҳида тус бериш учун қўлланилади. Одатда биттаси ишлатилган жадвалнинг ичида қолган иккитаси ҳам қўлланилиши кутилади. Бу тегларнинг қулайлиги шундаки, уларнинг ичидаги ҳар бир катак учун алоҳида ҳусусиятлар бериш талаб қилинмайди. Агар CSS ёрдамида бир маротаба ушбу тегларга жилолар биринтирилса, жадвал тарзда бир қанча ҳисботлар чиқараётганимизда уларнинг барчаси бирдай дизайнда сарлавҳа, юқори, қуи ва асосий қисми ажраган ҳолда бежирим намоён бўлади.

4.3. Жадвални шаклантириш

Жадвални яратишида энг муҳим қадам, ундаги сатрларни аниқлаб олишда. Барча жадвалдаги ката克拉р маълум аниқланган сатрда жойлашади. Жадвалдаги сатрни `tr` теги ёрдамида аниқланади. У ҳам юқорида қайд этиб ўтилган қийматлар қабул қилувчи қуидаги аломатларга эга: `align`, `bgcolor`, `bordercolor`, `char`, `charoff`, `valign`.

Таъкидлаб ўтиш жоизки, сатрнинг ҳусусиятларига нисбатан унда жойлашган ката克拉рнинг ҳусусиятлари устивор аҳамиятга эга. Яъни, мисол учун, битта қаторга `tr` аломатлари орқали умумий ранг берилсаю ундаги ката克拉рнинг ҳар бирига ўзига ҳос бошқа ранг берилса, ката克нинг ранги акслантирилади.

Жадвалдаги ката克拉р `td` теги ёрдамида ифодаланади ва улар албатта `tr` теги ичидаги жойлашиши шарт. Шунга қараб, ката克нинг қайси қаторда акслантирилиши аниқланади. Жадвалдаги барча кўринадиган объекtlар фақат ва фақат ката克 ичига ёзилади. Бир қатор тугагандан сўнг, кейинги қатор бошлангунча нимадир ёзиб уни ўша сатрлар орасида аксланишини кутиш керак эмас. Бу «`table`» тегининг синтаксисига зид ва биз билганимиздай шу каби хато кодлар браузерлар томонидан эътиборсиз қолдирилади ёки саҳифанинг бошқа жойида намоён бўлади.

Энг содда, фақат битта ката克дан иборат жадвал чизиш учун ҳам камидаги учта тег ичма-ич қуидаги тартибда ёзилиши шарт:

```
<table><tr><td>объект</td></tr></table>
```

Аслида фақат битта ката克дан иборат бўлса, жадвал ўрнига `div` теги ишлатилади. `Table` теги ёрдамида эса икки ўлчамли жадвалларни тушунарли, кўзимиз ўргангандаги тарзда акслантириш учун фойдаланилади. Демак, `td` теги жадвал элементининг асоси ҳисобланади. У қиймати бутун сон бўлган қуидаги аломатларга эга:

- **colspan** — Битта сатрдаги бир неча катақни бирлаштиради.

- **rowspan** — Бир устундаги кетма-кет катақларни бирлаштиради.

Булардан ташқари CSS ёрдамида ифодаласа бўладиган, эскирган дея ҳисобланувчи, тавсифлари юқорида келтирилган ушбу омилларга ҳам эга: **align**, **background**, **bgcolor**, **bordercolor**, **char**, **charoff**, **height**, **nowrap** (ёзув фақат битта сатрда чиқишини таъминлаш учун), **valign**, **width**.

Яна деярли ишлатилмайдиган ва браузерда аксланмайдиган катақ ичидаги маълумотларга изоҳ ёзиш ва уларни ўзаро ёки сарлавҳа билан боғлаш учун мўлжалланган қуидидаги эскирган теглар ҳам мавжуд: **abbr**, **axis**, **headers**, **scope**. Буларни қўллаш самарасиз ва тавсия этилмагани боис, ҳар бирига алоҳида тўхтамаймиз. Агар веб-саҳифа маълум дастурий таъминот ёрдамида таҳлил қилинса, катақлардаги маъмумотларни гуруҳлаш учун ушбу омиллар керак бўлиши мумкин.

Агар катақ ичидаги матн сарлавҳа сифатида намоён бўлиши керак бўлса, бошқаларидан ажраб туриши учун одатда сатрларни катақ ўртасига қалин ёзувда жойлаштирилади. Бунинг учун алоҳида аломатлардан фойдаланиб ётмасдан шунчаки **td** ўрнига **th** тегини қўллаш кифоя.

Жадвални яратишдан олдин унда нечта катақ бўлишини ва қандай тарзда жойлашишини аниқлаб олиш лозим. Ҳар бир сатрдаги катақларнинг умумий сони бир хил бўлмаса, браузер кутилгандагидек акслантирмайди. Агар бизга ҳамма катақ миқдори бир хил бўлмаган жадвал чизиши керак бўлса, аввал бир хил сондаги катақлардан иборат жадвал тузиб, кейин **colspan** ва **rowspan**лар ёрдамида бир неча катақларни умумлаштиришга тўғри келади. (Умуман олганда жадвал чизишидан аввал Microsoft Excel дастурини тасаввурга келтириш лойиҳалаштиришда яхши самара беради.)

4.4. Жадвал чизиши

Жадвал чизишида ҳар бир катаги ўзининг алоҳида чегараний чизиқларига эга эканлигини унугтаслик лозим. Яъни катаклар орасидаги ёки четки катак билан жадвал чегараси орасидаги чизиқлар аслида икки қатламдан иборат. Бу «cellspacing»га нолдан катта қиймат берилганда аниқ кўришади. Агар икки катакни умумлаштиргмаган ҳолда, фақатгина уларнинг орасидаги чегара кўринмаслигини истасак, ҳар икки катакнинг мос чегаравий чизиқларини яширишга тұғри келади. Бунга мисолни CSSга бағишиланган бўлимда кўриб ўтамиш. Браузерларнинг олдинги талқинларида агар катак ичидаги кўринадиган обьект бўлмаса, унинг чегараси чизилмас эди, бу эса жадвалнинг анчагина хунук акслантиради. Шунинг учун эски браузерларда чиқиши лозим бўлган бўш катаклар ичига «кўринувчи бўшлик» () кўйиш одат тусига кирганди. Янги талқиндаги браузерлар учун бу шарт эмас, фақат саҳифа эски браузерлар билан ишлашга ҳам мўлжалланган бўлса, буни инобатта олиш мақсади мувофиқдир.

Ушбу бўлимда танишган билимларимизни мустаҳкамлаш учун, мисол тариқасида битта жадвални кўриб ўтайлик. Аввал кўз олдимизга келтириб олишимиз учун жадвалнинг ўзини чизамиш:

№	Тур	Махсулот	Дона	Вазн
1.	Мева	Олма	120 та	23,1 кг
2.		Нок	60 та	13,4 кг
3.		Бехи	36 та	17,6 кг
4.		Анор	24 та	11,2 кг
5.	Полиз	Қовун	4 та	22,6 кг
6.		Тарвуз	3 та	19,7 кг
7.		Ҳандалак	10 та	22,8 кг
Умумий:			257 та	130,4 кг

Энди ҳар бир устун, сатр ва катакларга алоҳида эътибор қаратган ҳолда ушбу жадвални ифодаловчи HTML кодини ёзамиш:

```





```

```

<tr>
    <td>4.</td>
    <td>Анор</td>
    <td>24 ма</td>
    <td>11,2 кг</td>
</tr>
<tr>
    <td>5.</td>
    <td rowspan="3" valign="top">Полиэ</td>
    <td>Көсүн</td>
    <td>4 ма</td>
    <td>22,6 кг</td>
</tr>
<tr>
    <td>6.</td>
    <td>Тарбаз</td>
    <td>3 ма</td>
    <td>19,7 кг</td>
</tr>
<tr>
    <td>7.</td>
    <td>Хандалак</td>
    <td>10 ма</td>
    <td>22,8 кг</td>
</tr>
</tbody>
</table>

```

Эътибор қилиш керакки, «Мева» ёзилган катақда rowspan="4" қиймат берилган. Бу унинг тўртта одатий сатрга жойлашишини англатади. Яъни кейинги қаторларда камидা учта tr қатнашиши керак ва уларда шу катак тагига тўғри келувчи «td»лар ёзилмайди. Шунинг учун «Мева» ёзилган қатордаги «tr» ичидаги бешта «td», унинг остидаги «tr»ларда тўрттадан «td». «Умумий» сўзи ёзилган «tr» ичидаги эса, бор-йўғи учта «td». Сабаби биринчи катак учтасининг жойини эгаллаяпти.

«thead», «tfoot», «tbody» ва «col»ларни ишлатиш катақларни гурухларга ажратиб, уларга бир хил тус беришда жуда қулайлик келтиришини ҳам мисолда яққол қўрдик. Яна эътибор бериш керакки, «tfoot» теги «table»нинг қайси қисмида ёзилмасин, ундаги сатр жадвалнинг энг қуйида акслантирилади.

V БҮЛІМ

МЕДИА ФАЙЛЛАРНИ БОГЛАШ



5.1. Харитали расм қўйиш

HTML бошиданоқ матн ва расмни акслантиришга қаратилган, шунинг учун илк талқинлариданоқ расм қўйиш учун маҳсус тег киритилган. Агар расм бир неча қисмлардан ташкил топган харита сифатида намоён бўлса ва ундаги ҳар бир қисмга босгандга алоҳида манзилга ўтиш ёки амал бажаришга тўғри келса, аввал расмни маҳсус шакллар асосида объектларга тақсимлаб олиш лозим бўлади. Бунинг учун **map** тегидан фойдаланилади:

```
<map name="номи">майдонлар</map>
```

Бунда «номи» идентификатор бўлиб, битта саҳифа учун такрорланмас бўлиши талаб қилинади, «майдонлар» эса агеа теги ёрдамида кўрсатилади. Расмнинг маълум бўлагиги кўрсатувчи **area** теги қўйидаги аломатларга эга:

▪ **alt** — расм кўринмаганда акслантирилувчи муқобил матн.

▪ **shape** — ажратиладиган майдон шакли. Қиймати — аникланган тур:

- **circle** — доира.
- **default** — расмни тўлалигича қамраб олади.
- **poly** — кўпбурчак.
- **rect** — тўғри тўртбурчак, одатий қиймат.

▪ **coords** — математик декард координаталар тизими асосида расмга нисбатан унинг маълум қисмини ажратиб олиш учун кўрсатиладиган сонлар. Пиксел ҳисобида чап юқори бурчақдан бошлиб биринчи сон энига ва иккинчи сон бўйига нисбатан ҳисобланади. Агар «shape»да

айлана кўрсатилган бўлса, координаталар учта сондан иборат бўлиб, марказнинг эни ва бўйига нисбатан қаерда жойлашганини ҳамда радиусини англатади. Агар «shape»да тўғри тўртбурчак назарда тутилган бўлса, координата тўртта сондан иборат бўлиб, чап юқори ва ўнг қуий бурчак нуқталарини жойлаштан ўрнини кўрсатади. Кўпбурчакни ифодалаш учун унинг қирраларидағи нуқталарини соат мили йўналишига тескари тартибда жойлашиши бўйича координаталари кўрсатилади.

- **href** — майдонга «босилганда» қайси манзилга саҳифа йўналтирилишини кўрсатади, қиймати: ишорат.
- **hreflang** — «**href**» ёзилган хужжат қайси тилдалигини кўрсатади. Ушбу хужжатни акслантирувчи браузер ёки дастурний таъминот интерфейси қайси тилда бўлишини аниқлашга ёрдам беради.
- **nohref** — бу эскирган тег, «**href**» ўз вазифасини бажармаслигини таъкидлаш учун кўлланилган. HTML5 да «**href**» ёзилмаса «**nohref**» кўрсатилгандагидек ишлайди.
- **target** — «**href**» ёзилган хужжат қаерга юкланишини кўрсатади. [2.4.]
- **type** — «**href**» ёзилган хужжатта ММЕ тур кўрсатади. [2.4.]

Расм қўйишга мўлжалланган **img** тегининг қуидаги аломатлари мавжуд:

- **align** — расм аксланиши керак бўлган ташқи объектнинг майдонига қандай тарзда жойлашишини аниқлайди. Қиймати аниқланган тур бўлиб, матн билан бирга келган расмни бир-бирига нисбатан қандай кўринишда бўлишини ҳам аниқлайди. Бу эскирган аломат ҳисобланса ҳам амалда кўп кўлланилади, шунинг учун қийматларини батафсил кўриб чиқамиз:
 - **bottom** — расмнинг қуий матннинг остига текисланади.
 - **top** — расмнинг юқори қисми матннинг тепасига текисланади.
 - **left** — расм майдоннинг чап қисмiga текисланади.
 - **middle** — расмнинг ўртаси матн сатрининг ўртасига текисланади.

- **right** – расмни майдоннинг ўнг қисмига текислаши.
 - **alt** – агар кўрсатилган манзилдаги расмни аксланириш имкони бўлмаса, ушбу аломатдаги матн намоён бўлади. Қиймати: сатр.
 - **border** – расм атрофидаги ҳошия қалинлиги (эскирган).
 - **height** – расм баландлиги.
 - **width** – расм эни.
 - **hspace** – расмнинг ён томонларидағи қолдирилладиган оралик.
 - **vspace** – расмнинг тепа ва пастидан қолдирилладиган бўшлиқ.
 - **src** – расм файлининг манзили, қиймати: ишорат.
 - **srcset** – ушбу аломат HTML5да қўшилган бўлиб, браузерга келтирилган рўйхатдаги маъкул расмни намоён қилишга имкон яратади. Масалан, айни веб саҳифа компьютерда очилганда бошқа расм, планшетда очилганда бошқа ва телефонда очилганда бошқа ўзига мосини ундаги браузер аниқлаб кўрсатаверади. Ҳозирча ушбу ажойиб имкониятни «Internet Explorer» ва «Android»нинг ўз браузери кўлламаслигини ҳам таъкидлаб ўтиш жоиз. Бунда рўйхат вергуллар билан ажратилган расмларнинг манзил ва ўлчамларидан иборат бўлади. Ўлчам манзилдан бўш жой билан ажратилган бўлиб, эни неча пиксел бўлишини ёки расм неча марта катталаштирилиши лозимлигини кўрсатади. Агар сондан кейин «w» қўйилса энини, «x» қўйилса қанча марта катталаштиришини англатади.
 - **sizes** – ўзгарувчан ўлчамли саҳифалар учун расмнинг қандай оралиқда кенгайиш ёки торайиш мумкинлигини аниқлайди. Ташқи объектга нисбатан фоизда қиймат бериш учун «vw» ҳарфлар биримасидан фойдаланилади, «%» белгиси ишлатилмайди. Масалан: sizes="100vw". Қиймат сифатида ҳисобловчи функциядан [8.4.10.] ҳам фойдаланиш мумкин:

```
<img srcset="a1.jpg 2x,a2.png 150w,a3.gif 200w"  
      sizes="(max-width: 300px) 100px, (min-width: 50em)  
            50vw,calc(33vw-100px)">
```

▪ **usemap** – расмнинг бўлакларидан ҳарита сифатида фойдаланиш учун кўлланилади. Юқорида кўриб чиқилган **map** теги ёрдамида боғланади:

```
<map name="shakllar">  
    <area shape="circle" href="circle.htm"  
          coords="320,120,100" title="Доира">  
    <area shape="poly" href="poly.htm"  
          coords="9,4,13,0,79,0,7,24" title="Майдон">  
    <area shape="rect" href="rect.htm"  
          coords="60,20,260,420" title="Тўртбурчак">  
</map>  

```

5.2. Товушли файл қўйиш

HTML5нинг энг катта ютуқларидан бири товушли ва лавҳали файлларни ўзи тезкор ижро этишни таъминлайди. Ҳозирда барча браузерларнинг юқори талқинлари ушбу тегни тўлақонли намоён этади.

Товушли файлни ижро этиш учун **audio** тегидан фойдаланилади. У қуйидаги аломатларга эга:

▪ **autoplay** – саҳифа тўла юкланиб бўлиши билан товуш ижро этилишини таъминлайди. Зарурмас қиймат.

▪ **controls** – ижрони бошқариш (тўхташиш, қайта қўйиш, овоз баландлигини созлаш) тутмаларини кўринарли бўлишини таъминлайди. Зарурмас қиймат.

▫ **loop** – ижро этиб бўлингандан сўнг қайтадан бошлиди, зарурмас қиймат.

▫ **muted** – овоз ўчирилган ҳолда намоён бўлади, зарурмас қиймат.

▫ **src** – ижро этиладиган файл манзили, қиймати: ишорат.

Кейингй китобимизда JavaScript билан танишганимиз-

да, товушларни объектлар ва улар устида амал бажарылганда мос равища юклаш ва ижро этилишини таъминлаш хусусида батафсил түхталамиз. Бундай имканийтлар замонавий веб сайтынан яратышда жуда кенг күлланилади ва оммабоп саналади.

Мабодо қайсиидир браузер ушбу тегни акслантира олмаса, унинг ичидаги ёзувчи шундоқ күрсатади. Оммабоп браузерлар ичида бундайлари қолмаган, лекин ҳар эҳтимолга қарши файлни ижро этадиган дастурий таъминот мавжуд бўлмаган ҳоллар учун ҳам кўчириб олишга имкон берувчи изоҳ қолдириш мақсадга мувофиқ бўлади:

```
<audio autoplay loop src="music.ogg">  
    Уибу файлни ижро этишига имкон бўлмади  
    <a href="music.ogg">Марҳамат, юклаб олинг</a>  
</audio>
```

5.3. Лавҳали файл қўйиш

Энди замонавий браузерларда видео лавҳаларни томоша қилиш учун қўшимча дастурий таъминот ёки «FLASH» сингари кенгайтмаларни юклаш шарт эмас. HTML5 видео кўрсатиши барча стандарт воситалари билан ўз зиммасига олади. Бунинг учун алоҳида `video` теги ишлатилади. У қўйидаги аломатларга эга:

- **preload** — саҳифа юкланиб бўлиши билан лавҳа файлини қандай тарзда кўчириш лозимлигини кўрсатади. Киймати — аниқланган тур:

- **metadata** — фақат бош маълумот қисми (лавҳа ҳажми, биринчи кадр, давомийлик вақти ва бошқалар) юкланади.

- **none** — одатий қиймат, файлни юкламайди.
 - **auto** — лавҳа тўлалигича юкланиб олинади.

- **autoplay** — саҳифа тўла юкланиб бўлиши билан лавҳа намойиш этилишини таъминлайди. Бу аломат кўрсатилганда «preload» ишлатилмайди. Зарурмас қиймат.

- **controls** — ижрони бошқарий (тўхташиш, қайта қўйиш, овоз баландлиги созлаш) тутмаларини кўринарли бўлишини таъминлайди. Зарурмас қиймат.

▪ **loop** — ижро этиб бўлингандан сўнг қайтадан бошлиди, зарурмас қиймат.

▪ **poster** — лавҳа намойиш этилмасдан тасвир майдонида кўринадиган расмнинг манзили, қиймати: ишорат. Агар бунда манзил кўрсатилмаса ёки кўрсатилган расми очиш имкони бўлмаса, тасвир майдонида биринчи кадр акслантирилади.

▪ **src** — ижро этиладиган файл манзили, қиймати: ишорат.

▪ **height** — лавҳа намоён бўладиган майдон баландлиги, қиймати сон ёки фоиз. Агар қиймат берилмаса, лавҳа хусусиятларида кўрсатилган баландлик олинади, уни ҳам аниқлаш имконсиз бўлса «**poster**»даги расм ўлчами-даги бўйи олинади. Расм ҳам мавжуд бўлмаганда 150 пиксел баландлиқда кўрсатилади.

▪ **width** — лавҳа намоён бўладиган майдон эни, қиймати сон ёки фоиз. Одатий қиймати «**height**»ники сингари аниқлаб бўлмаса, 300 пиксел кенглиқ ўрнатилади.

Ушбу тегда тасвирнинг ўлчами қатъий берилмаса, лавҳанинг асл ҳажми қандай бўлса, шундай тақсимот билан намоён этилади. Албатта бу доим ҳам самарали на-тижа бермайди. Эни ва бўйи қатъий белгиланганда унда ижро этилаётган лавҳа баландлиги ёки кенглиги бўйича унга мослашади, аммо пропорционаллик сақланиб қолади. Яъни кадрлар чўзилиб ёки сиқилиб қолмайди.

5.4. Танланган файлни юклаш

Албатта файл юкланаар экан, унинг ҳажмига ҳам эътибор бериш лозим, интернетда ҳар бир байт ҳажм ўз ҳисоб-китобига эга. Шунинг учун медиа (расм, товуш, лавҳа) файлларнинг ҳам ҳажми кичик аммо ўз сифатини йўқотмаганидан фойдаланиш тавсия этилади. Барча тоифада медиа файлларни ҳар доим ҳам ҳамма браузерлар кўллайвермайди. Баъзан маҳсус кенгайтмалар ўрнатишга тўғри келади. Фойдаланувчиларга маълум дастурларни юкламасанг, бизнинг веб-саҳифамизни тўла қонли кўра олмайсан, дея дакки бериш ножойиз. Фойдаланувчилар

веб-саҳифани қандай браузер орқали очишмасин, доимо у тўлиқ имкониятини намоён эта олиши керак.

Масалан, энг сифатли ва ихчам товушли файл «ogg» (кенгайтмалиси) ҳисобланади. Аммо «Internet Explorer» ва «Safari» бундай турдаги файлни қўлламайди. Бу икки браузер учун «AAC» ёки «MP3» шаклида, қолганларга ҳажми кичик «ogg» кўринишидаги файлни юкла деб HTMLнинг ўзида қўшимча текширув шартларини қўймасдан ёзиш учун **source** тегидан фойдаланилади. Бу тег ёрдамида файлнинг бир неча мұқобил манзили келтирилади. Браузер ўзи қўллайдиган биринчи файлни намойиш этади. Ў қуйидаги аломатларга эга:

- **media** — қурилма тури. [2.4.]
- **src** — файл манзили, ишорат.
- **srcset** — ишоратлар рўйхати. [5.1.]
- **sizes** — ўлчамлар рўйхати. [5.1.]
- **type** — файлнинг МІМЕ тури. [2.4.]

Ушбу тег «audio» ёки «video» теглари ичида ёзилганда, уларнинг **src** аломатига қўймат бериш шарт эмас, ички «source» тегларининг «src» аломатларида кўрсатилган рўйхатдан биринчи ўзи тушунган файлни ижро этаверади. Мисол:

```
<video width="400" height="300" controls="controls">
    <source src="lavha.ogv"
        type='video/ogg; codecs="theora, vorbis"'>
    <source src="lavha.mp4"
        type='video/mp4; codecs="avc1.42E01E,mp4a.40.2"'>
    <source src="lavha.webm"
        type='video/webm; codecs="vp8, vorbis"'>
</video>
```

Расмлар ҳам баъзан вектор сифатида, баъзан ҳаракатли «gif» ёки бошқа шаклда бериладиган бўлса, ҳамма браузерларнинг ҳам тўлиқ қўллашига кафолат йўқ. Бундай ҳоллар ҳам мұқобил файл сифатида оммабоп расм шаклини «source» теги ёрдамида кўрсатиш мумкин. Муаммо шундаки, «audio» ва «video» теглари HTML5да киритилган бўлиб, уларнинг очилувчи ва ёпилувчи қисмлари мавжуд ва ичида «source» тегини

қўлламайдиган браузерларда акслантириладиган матни ёзиш имконияти яратилган. Лекин «*img*» HTMLнинг биринчи талқинидаёқ киритилган ва унинг ёпувчи қисми мавжуд эмас. Бу ҳолда қўшимча равишда киритилган *picture* тегидан фойдаланилади:

```
<picture>
  <source srcset="tasvir.svg">
  
</picture>
```

Айни бир сахифа экраннинг тури, ўлчами, тик ёки ётиқлигига қараб турли суратлар ичидан мос расмни танлаб қўрсатишда бундай усул яхши самара беради.

Ижро этилаётган лавҳа ёки товушга қўшимча изоҳ матнини қўрсатиш жоиз бўлса, унда *track* тегидан фойдаланилади. Айниқса кино намойиш бўлаётганда қаҳрамонларнинг гапларини айни пайтда ёзув сифатида қўрсатиб туриш кўп ҳолларда талаб этилади. Бундай имкониятни тақдим этиш учун «*audio*» ёки «*video*» теглари ичига *track* теги ёзилади, у маҳсус тавсифномали матн (субтитр) ёзилган файлга мурожаат қиласи. Унда ёзилган сатрларни қўрсатилган вақти бўйича экранда намойиш этади.

Ушбу *track* (йўлакча) теги қуйидаги аломатларга эга:

- **kind** — йўлакчанинг турини қўрсатади, қиймати аниқланган тур:

- **subtitles** — тавсифномали матн лавҳанинг устига чиқади. Бунда ҳар бир айтилаётган гапнинг ёзуви ўша пайтда акслантирилади. Бошқа тилда ёзувни ўқиб бормоқчи бўлганларга ёки талаффуз этилаётган сўзларнинг ёзувда қандай ифодаланишини ўрганмоқчи бўлганларга, умуман олганда кулоғи эшитмайдиганларга бу жуда ҳам қулагай восита.

- **captions** — бунда тўлиқ барча гап ёзилмайди, айрим қисмларга, айтайлик бошқа тилда гаплашган вақтда ёки қўшиқ куйлананаётганда унинг матни қўринади, холос.

- **descriptions** — бу одатда экрандаги воқеиликка изоҳ берилаётганда, яъни фақат қўшимча фикр ёзув тарзда аксланиши лозим бўлганда қўлланилади.

- **chapters** — лавҳанинг айни пайтда қайси мавзуга

оид қисми ёки бўлими ижро этилаётганини кўрсатиб туриш учун ишлатилади.

◦ **metadata** – бунда матн лавҳада кўрсатилмайди. Махсус қўшимча кодлар ёрдамида таҳлил қилиш учунгина фойдаланилади.

▪ **src** – файлнинг манзили. Тўлиқ ёки нисбий ишорат.

▪ **srlang** – йўлакчанинг тили. [2.7.]

▪ **label** – йўлакчанинг номи, агар у кўрсатилмаса созлашлар панелида «untitledN» сифатида кўринади. Шунинг учун бу аломатта сатр сифатидаги қаймат бериш тавсия этилади.

▪ **default** – зарурмас қайматга эга бу аломат битта лавҳа ёки товуш теги ичида қатнашган йўлакчаларнинг фақат биттасига ёзилади ва одатда айнан ушбу йўлак фойдаланишини англаради.

Ушбу бўлимда олган билимларимизни мустахкамлаш учун битта мукаммалроқ мисол кўриб ўтамиш:

```
<video width="500" height="400" controls>
    <track kind="subtitles" src="lavha.en.vtt"
        srclang="en" label="English">
    <track kind="subtitles" src="lavha.uz.vtt"
        srclang="uz" label="Ӯzbekcha" default>
    <track kind="subtitles" src="lavha.ru.vtt"
        srclang="ru" label="Русский">
    <source type='video/ogg; codecs="theora, vorbis"'
        src="video/lavha.ogv">
    <source type='video/mp4; codecs="avc1.42E01E,mp4a.40.2"'
        src="video/lavha.mp4">
    <source type='video/webm; codecs="vp8, vorbis"'
        src="video/lavha.webm">
    «video» теги ишламади! HTML5ни қўллайдиган
    оммавий браузердан фойдаланинг!
</video>
```

5.5. Ностандарт медиа қўйиш

Баъзан шундай медиа файлларни акслантиришга тўғри келади-ки, уларнинг форматини оммабоп браузерлар тушишмайди. Намойиш этиш учун «кўшилгич» (плагин) деб аталувчи маҳсус қўшимча дастурлардан фойдаланишга тўғри келади. Ёки алоҳида дастурий кодлар ёзиш талаб этилади. Браузер қўллайдиган дастурий кодларни «скрипт» деб юритамиз. Улар маҳсус **script** теги ичидаги ёзилади. Кейинги китобда айнан «JavaScript» билан батафсил танишамиз. Ҳозирча ушбу китобда баён этилган кодларни намойиш этиш учун скриптни қўлламайдиган ёки созлашлар бўлимидан бундай имконияти ўчирилган браузерлар ҳам кифоя қилади. Скрипт браузерда қўлланилишини текшириш учун **noscript** теги ишлатилади. Агар унинг ичидаги ёзилган матн веб-саҳифада акслантирилса, демак, браузерда ёзилган скриптлар ишламайди.

Браузерлар ва уларга қўшилгичлар қандайлигига қараб, юкланаётган медиа файлнинг кўриниши турлича бўлиши мумкин. Уларни **embed** теги ёрдамида биректирилади ва куйидаги аломатлар ёрдамида созлаш мумкин:

- **height** – объект баландлиги, қиймати: ўлчам;
- **hidden** – объектни кўринмаслигини таъминлайди, мантиқий қиймат;
- **pluginspage** – «кўшилгич»ни қаердан юклаб олишни кўрсатади, қиймати: ишорат;
- **src** – файл манзили, қиймати: ишорат;
- **type** – файл MIME-тури;
- **width** – объектнинг эни, қиймати: ўлчам;

Мисол:

```
<embed width="400" height="300" src="/flash/anim.swf"
       type="application/x-shockwave-flash"
       pluginspage="https://get.adobe.com/flashplayer">
```

5.6. Объектларни бириктириш

Медиа файл қўйилганда, унга ёзиладиган изоҳлар ва қўшимча теглар орқали боғланган элементлар ягона объект сифатида мужассамлаштириб акслантирилиши лозим бўлади. Яъни расм ёки бошқа медиа файл бир жойдан бошқа ерга қўчирилишига тўғри келса, уни тавсифловчи матн ва бошқа элементлар умумий қўриниши ўзгармаган ҳолда биргаликда силжиши керак. Шундай вазиятларда бир неча теглар орқали яратилган элементларни бириктириш учун уларни HTML5да янги киритилган **figure** теги ичидаги ёзиш талаб этилади. Ушбу мажмууга изоҳ **figcaption** теги ёрдамида қўйилади, фақат у «**figure**» теги ичидаги биринчи ёки охирги элемент бўлиши шарт. Мисол:

```
<figure>
  <p></p>
  <hr>
  <audio autoplay loop src="kuy.mp3"></audio>
  <figcaption>Эски шаҳар</figcaption>
</figure>
```

5.7. Ташқи HTML файлни акслантириш

Бутунлай бошқа, ўзига хос «CSS» ва «JavaScript» ларига эга ташқи HTML файлни саҳифанинг бир қисмида наамоён этиш учун **iframe** тегидан фойдаланилади. Имкони борича уни ишлатмаслик тавсия этилади, аммо баъзан бошқа манъбадаги маълумотларни (муаллифлик хуқуқини бузмаган ҳолда) биргаликда акслантиришга тўғри келган вазиятда фрейм қўллашга мажбур бўлинади. Ушбу тег қуидаги аломатларга эга:

- **allowfullscreen** — фреймни тўлиқ экранда аксланишини таъминлайди (зарурмас қиймат);
- **allowtransparency** — фреймнинг ортрангини шаффоғ қиласди (зарурмас қиймат);
- **frameborder** — фрейм атрофида чегаравий чизик бўлишини аниқлайди (мантиқий қиймат);

- **height** – фрейм баландлиги;
- **marginheight** – фрейм тела ва пастидаги ҳошия қалинлиги (ўлчам);
- **marginwidth** – фрейм ён томонларидағи ҳошия қалинлиги (ўлчам);
- **name** – фрейм номи;
- **sandbox** – фреймдаги саҳифага чеклов ўрнатади. Масалан, сиз ижтимоий тармоқнинг қайсиdir имкониятини сайтингизга улаш учун фрейм орқали боғладингиз, лекин маълум бўлишича, сиз улашишни истамайдиган маълумотларни ҳам олишга унинг имконияти мавжуд. Шундай ҳолларда чекловлар қўйишга эҳтиёж сезилади. HTML5да қўшилган ушбу алломатга қиймат кўрсатилмаган тақдирда, барча чекловлар ўрнатилган деб ҳисобланади. Куйидаги қийматларни (бир нечаси берилса, бўш жой билан ажратилади) кўрсатиш билан, маълум рухсатлар бериш мумкин:
 - **allow-forms** – форма ёрдамида қийматларни юборишига;
 - **allow-modals** – фрейм ичидан модал ойна [10.1.] очишига;
 - **allow-orientation-lock** – экранни қулфлашни тақиқлашига;
 - **allow-pointer-lock** – Pointer Lock API (сичқонча ҳаракатларини назорат қилувчи скрипт)дан фойдаланишига;.
 - **allow-popups** – сузуб чиқувчи («window.open», «target=_blank», «showModalDialog»лар ёрдамида яратилган) ойналар кўринишига;
 - **allow-presentation** – Presentation API (ташқи мониторларни қўлловчи скрипт)ни ишлатишига;
 - **allow-same-origin** – сузуб чиқувчи ойнани тақиқлаб, фақат ташқи саҳифа манбасидан олинган ишорат бўйича юклашга;
 - **allow-scripts** – сузуб чиқувчи ойна яратишини чеклаган ҳолда скрипtlарни ишлашига;
 - **allow-top-navigation** – ишоратларни ташқи саҳифада очишига;
- **scrolling** – айлантиргич қўйилишини аниқлайди.

Қиймати: «auto», «yes», «no»лардан бири.

- **seamless** — фреймни ташқи саҳифанинг қисми сифатида акслантириш (зарурмас қиймат).
- **src** — фреймга юкланувчи саҳифа манзили, қиймати ишорат.
- **srcdoc** — HTML код. Саҳифа манзилини кўрсатиш ўрнига унда бўлиши лозим бўлган кодни ушбу аломат орқали бериш мумкин. Ушбу аломат HTML5да қўшилган ва уни «IE» ва «Edge»лар қўлламаслиги мумкин. Агар код ёзилса, «src»аломатидаги манзил эътиборсиз қолдирилиб, ушбу код намойиш этилади.
- **width** — фрейм эни.

Мисол:

```
<iframe  
    width="200"  
    height="100"  
    frameborder="1"  
    scrolling="no"  
    srcdoc="<!DOCTYPE html>  
        <div>Бу оддий DIV эмас</div>">  
</iframe>
```

VI БҮЛİM

ГРАФИК ВА ТАСВИРЛАР ЧИЗИШ



6.1. CANVAS теги

HTML5 имкониятларидан фойдаланиб, браузерда исталган тасвири чизиш мумкин. Тайёр суратдан фойдаланишдан расм чизишнинг кулигиги, браузердаги ҳаракатга мутаносиб равишда чизиладиган тасвирнинг ўзгариб боришидир. Расм чизиладиган майдон canvas теги билан белгиланади. Унга тасвир эса маҳсус скрипtlар ёрдамида чизилади. Кейинги «JavaScript»га бағищланган китобимиизда суратлар чизиш, уларни ўзимиз истагандай ҳаракатга келтириш ҳақида батафсил тўхталиб ўтамиз. Ҳозирча унинг блокли тег эканлигини ва нима учун қўлланилишини билишимиз кифоя. Унинг бор-йўғи иккита ўзига ҳос аломати мавжуд:

- height – баландлиги, одатий қиймати 150px.
- width – кенглиги, одатий қиймати 300px.

Тайёр расмни қўймасдан ёки маҳсус скрипtlарни ишлатмасдан HTML5да тасвир ҳосил қўлиш мумкин эмасми, деган ҳақли савол туғилиши табиий. Албаттa мумкин.

6.2. SVG теги

SVG (Scalable Vector Graphics) – XML шаклидаги векторли ўзгарувчан графикни тасвирлаш учун қўлланилади. У қолип сифатида canvas тегига ўхшаса ҳам, фарқи шундаки, унинг ичидаги маҳсус теглар ёрдамида шакллар чизиш мумкин. SVG бошқа соҳаларда ҳам кенг ишлатиладиган стандарт бўлиб, энг муҳим жиҳати шундаки, расмнинг ҳажми ўзгарганда ҳам сифати юқориilihiga

қолаверади. Сабаби унда XML шаклида ёзилган кодлар таҳлил қилиниб, браузернинг ўзи тасвири ҳосил қиласди. SVG тегининг ичида геометрик шакллар чизишга мўлжалланган маҳсус теглар ишлатилади.

Умуман олганда SVGни алоҳида XML тил сифатида қараш мумкин, унинг ичида кўпгина HTML теглар ўзига хос равищда фойдаланилади. Биз унинг асосини ўрганиб, фақат у ёрдамида икки ўлчамли Декарт координаталар тизимида стандарт шаклларни маҳсус ҳусусиятларини белгилаган ҳолда чизиш ва ҳаракатлаштиришни кўриб ўтамиз. SVG теги қўйидаги аломатларга эга:

- **version** — талқини, ҳақиқий сон. Қайси қоидалар асосида чизиш кераклигини кўрсатади.

- **baseProfile** — SVG қонуниятлар мажмуасининг ихчам жамланмасини кўрсатади, бу тасвирининг тўғри аксланишини таъминлаш учун зарур. Доим ҳам ҳамма қонуниятларни тўлалигича юклаб олиб фойдаланиш яхши натижа беравермайди. Қиймати аниқланган тур:

```
baseProfile = "none" | "full" | "basic" | "tiny"
```

Уяли қурилмалар учун ўзак (basic) ёки ихчам (tiny) мажмуани танлаш тавсия этилади.

- **width** — ички майдонининг эни. Қиймати сон ёки фоиз. Агар унга нол берилса, ундаги тасвиirlар чизилмайди.

- **height** — ички майдонининг бўйи. Ҳусусияти эни-ники сингари.

- **viewbox** — майдоннинг ичига шаклнинг қандай масштабда жойланишини аниқлайди. Агар ушбу аломат кўрсатилмаса ёки унга «none» қиймат берилса, тўла майдонда ўлчами ўзгармаган шакл акслантирилади. Қиймати сатр бўлиб, унда бўш жой ёки вергул билан ажратилган тўртта бутун сон қатнашиши керак. Мос равищда «X» ва «Y» бўйича бошлиғиҷ чегара (яъни тасвирининг шунча нуқтагача бўлган қисми кесиб юборилади), ҳамда эни ва бўйи бўйича ўлчамни ўзгартириш фоизи.

- **preserveAspectRatio** — маёштабни ўзгартирган ҳолда мослаштириш. Тасвирининг ўлчами «viewbox»да кўрсатилган қолип бўйича майдонга биз хоҳлагандай жойлаш-

маган тақдирда унинг жойини ва ҳажмини ўзгартириш учун ушбу аломатдан фойдаланилади. Қиймати сатр, учта қисмдан иборат. Агар SVGнинг ичида тасвир мавжуд бўлса, жойлаштиришни айнан унинг масштабига қараб амалга ошириш учун биринчи сўз ўрнига «defer» ёзилиши мумкин. Одатда биринчи сўз ишлатилмайди ва иккитаси билан кифояланилади. Асосий қийматда тасвир майдонининг (тўғри тўртбурчак шаклида) бир уни ёки маркази қаерда туришини кўрсатилади: қуйидагича шаблонда ёзилади:

X[Min Mid Max]	Y[Min Mid Max]	мослиги
----------------------	----------------------	---------

Бунда мослиги қуйидаги қийматлардан бирини қабул қилиши мумкин:

- meet – пропорция сақланиб, тасвир тўлалигича жойланади, одатий қиймат.

- slice – пропорцияни сақлаган ҳолда майдонни тўлдиришга уринади, тасвир тик ёки уфқий кесилиши мумкин.

- none – пропорцияни сақламай майдонни тўлдиради (тасвир бузилади).

Мисол:

```
<svg viewBox="0 0 300 300"
      preserveAspectRatio="XMinYMax meet">
    . . .
</svg>
```

- snapshotTime – ҳаракатланувчи суратларда қайси кадамдаги тасвирни акслантириш лозимлигини кўрсатади, animateColor сингари теглар ишлатилганда фойдаланилади. Афсус, ҳозирча браузерлар буни қўлламайди.

- playbackOrder – ўз ичидағи элементларга мурожаат қилишда уларни излаб топиш йўналишини кўрсатади. Қиймати аниқланган тур:

- forwardOnly – кетма-кетликда фақат олдинга қараб излайди.

- all – иккала йўлалишда ҳам излашни таъминлайди.

- **timelineBegin** — ҳаракатли тасвир бўлса, у қачон на-
мойиш бўлишини кўрсатади. Қиймати: аниқланган тур:
 - **onLoad** – саҳифа юкланиши билан, одатий қиймат;
 - **onStart** – кўрсатилган вақт ҳаракати бошланганда;
- **contentScriptType** — саҳифанинг МИМЕ тури. [2.4.]
кўшимча скриптдан фойдаланилган бўлса, кўрсатилган
файлнинг тавсиф тури.
- **zoomAndPan** — фойдаланувчига тасвирнинг ҳажмини
ўзгартириш имконини (масалан лупа ёрдамида) яратади.
Қиймати: **magnify** ёки **disable**.
- **focusable** — фокуслаш (тугма билан бошқариш
учун). Қиймати: аниқланган тур:
 - **true** – фокус фаол бўлиши мумкин, «Tab» тугма-
сида бошқарилади.
 - **false** – фокус фаоллашмайди.
 - **auto** – агар ичида фокусланувчи элемент бўлмаса,
фаоллашмайди.

Ушбу технология очиқ кодда бўлгани боис ҳам кенг қўлланилади. Агар янги талқинлари чиқиб, тадбиқ этиладиган бўлса, «version» аломатида уни кўрсатиб қўйилаверилади. Авваллари бундай имкониятлар бўлмаганлиги сабабли, «Флеш» каби кўшимча дастурий таъминотдан фойдаланишга эҳтиёж туғиларди, у эса тасвирларни пайдо қилишда ўзи патентлаган ёпиқ коддан фойдаланади. Флеш ишлаши учун эса маҳсус дастур операцион тизимга ўрнатилиши ёки браузер учун кенгайтма юкланилиши талаб этилади. Мутахассислар бундай кўшимча дастур билан тасвирлар ҳосил қилишда компьютерларнинг кўп ресурслари сарфланилишини, айниқча электр энергиясини ортиқча исроф бўлишини, ҳамда техник қурилмалар зўриқишини исботлаб, HTML5да бутунлай флещдан воз кечишини маъқул тошишган. Флешнинг ноёб имкониятлари лавҳа ва товушларни ижро этиш ҳамда ҳаракатли тасвирларни пайдо этиш эди. Кўриб ўтганимиздай ҳозирда HTML5да бу имкониятларнинг барчаси мавжуд. Улар очиқ кодли, ҳамма учун мўлжалланган, ортиқча дастурий таъминот ва техник имкониятлар талаб қилмайди.

6.3. Геометрик шакллар қўйиш

SVG тегининг ичида бир неча хил геометрик шакллар ҳосил қилиш мумкин. Улар ўзига хос теглар ёрдамида ёзилиб, қуидаги умумий аломатларга эга:

- **stroke** — чегараловчи чизиқ ранги. Қиймати: ранг.
- **stroke-width** — чегараловчи чизиқ қалинлиги, пиксел ҳисобида, қиймати: сон.
- **stroke-linecap** — чизиқ қалинлаштирилганда йўғонлиги асл чизиқقا нисбатан қандай кентайишни аниқлайди. Қиймати аниқланган тур:
 - **butt** – фақат икки ёнига бир хилда энлик қўшилади.
 - **square** – ҳар тарафига бир хилда кентайиб боради.
 - **round** – чизиқнинг учлари ёйсимон тарзда «қиррасизлантирилади».
- **stroke-linejoin** — чизиқлар туташганда ҳосил бўла-диган бурчакнинг «ўткирлиги»ни кўрсатади. Қиймати аниқланган тур:
 - **miter** – учли (ўткир)
 - **bevel** – қиррали (ўткир учи кесилгандагицек иккита қирраси бор)
 - **round** – ёйсимон (ўтмас)
- **stroke-miterlimit** — агар «**stroke-linejoin="miter"**» қиймат берилган бўлса, ушбу аломатда кўрсатилган сон чизиқларнинг ички туташган нуқтаси билан четки чегараси орасидаги масофа энг кўпи билан қанча бўлишини белгилайди.
- **stroke-dasharray** — чегараловчи чизиқ узилишлари. Қиймати: сатр. Унда бир неча сонлар бўш жой билан ажратиб ёзилади ва улар мос равища тақорорий кетакет келувчи чизиқча ёки узилишнинг узунликларини аниқлайди.
- **stroke-dashoffset** — «**stroke-dasharray**»да кўрсатилган биринчи сон чегаравий чизиқнинг бошланғич чизиқчасининг узунлигини англатади. Агар биринчи чизиқча қисқа, қолганлари кўрсатилган бўйича аксланиши керак бўлса, ушбу аломат ёрдамида бошланғич чизиқчани кесилади. Қиймати: «**stroke-dasharray**»да кўрсатилган биринчи сондан кичкина бутун сон.

- **stroke-opacity** — чегаравий чизикнинг шаффофлиги. Қиймати 0 ва 1 орасидаги ҳақиқий сон. 0 – кўринмас даражада тиник, 1 – ўз рангида (шаффофлиги йўқ).

- **fill** — шаклнинг ички ранги, қиймати: ранг.
- **fill-opacity** — шакл ичининг шаффофлиги. Қиймати «stroke-opacity»нидиай.

Таъкидлаб ўтиш жоизки, SVG теги XML тилига мўлжалланганлиги учун, унинг ичида қўлланилган шакл чизувчи тегларнинг ёпувчи қисми мавжуд бўлмаса, уларни «*/>*» билан тугатиш талаб этилади. Акс ҳолда хатолик юзага келиб, кейинги шакл аксланмаслиги мумкин.

6.3.1. Тўғри тўртбурчак

Тўғри тўртбурчак **rect** теги ёрдамида қўйилади. У қуйидаги аломатларга эга:

- **x** — чап юқори нуқтанинг уфқий координатаси.
- **y** — чап юқори нуқтанинг тик координатаси.
- **width** — тўғри тўртбурчак эни.
- **height** — тўғри тўртбурчак бўйи.
- **rx** — қиррасининг x ўқи бўйича эгилиш радиуси.
- **ry** — қиррасининг y ўқи бўйича эгилиш радиуси.

Мисол:

```
<svg width="120" height="120" viewBox="0 0 120 120">
    <rect x="6" y="8" rx="8" ry="8" fill="aqua"
          width="92" height="98" stroke="red" />
</svg>
```

6.3.2. Доира

Доира **circle** теги ёрдамида чизилади. У қуйидаги аломатларга эга:

- **cx** — чап юқори нуқтанинг уфқий координатаси.
- **cy** — чап юқори нуқтанинг тик координатаси.
- **r** — x ўқи бўйича қиррасининг эгилиш радиуси.

Мисол:

```
<svg width="120" height="120" viewBox="0 0 120 120">
  <circle cx="60" cy="60" r="48" fill="aqua" stroke="blue"/>
</svg>
```

6.3.3. Эллипс

Эллипс (қовунсимон шакл) **ellipse** теги ёрдамида чизилади. У қуйидаги аломатларга эга:

- **cx** — марказининг уфқий координатаси.
- **cy** — марказининг тик координатаси.
- **rx** — x ўқи бўйича радиуси.
- **ry** — y ўқи бўйича радиуси.

Мисол:

```
<svg width="120" height="120" viewBox="0 0 120 120">
  <ellipse cx="60" cy="60" rx="60" ry="20"
    fill="aqua" stroke="blue" />
</svg>
```

6.3.4. Кесма

Кесма **line** теги ёрдамида чизилади. У қуйидаги аломатларга эга:

- **x1** — кесма бошининг x ўқи бўйича координатаси.
- **y1** — кесма бошининг y ўқи бўйича координатаси.
- **x2** — кесма охирининг x ўқи бўйича координатаси.
- **y2** — кесма охирининг y ўқи бўйича координатаси.

Мисол:

```
<svg width="120" height="120" viewBox="0 0 120 120">
  <line x1="1" y1="1" x2="119" y2="119" stroke="blue" />
  <line x1="20" y1="100" x2="100" y2="20" stroke="blue" />
</svg>
```

6.3.5. Синиқ чизик

Синиқ чизик **polyline** теги ёрдамида чизилади. Унинг **points** аломатида координаталар жуфтлиги вергул билан берилади.

лан, нұқталар эса бүш жой билан ажратиб сатр сифатыда берилади. Мисол:

```
<svg width="80" height="80" viewBox="0 0 80 80">
    <polyline points="40,0 80,80 0,40 40,0"
        fill="none" stroke="red" />
</svg>
```

6.3.6. Күпбурчак

Агар синиқ чизиккінг бошланғич нұқтасининг координатасини берилған сатр охирига ҳам құшилса, ёпиқ күпбурчак шакли ҳосил бўлади. Синиқ чизик билан күпбурчак ифодаланадиган бўлса, томонлари кесишиб қолишига ҳам имкон туғилиши мумкин, бунга йўл қўймаслик мақсадида, күпбурчак чизиш учун маҳсус **polygon** теги қўлланилади. Мисол:

```
<svg width="80" height="80" viewBox="0 0 80 80">
    <polygon points="40,0 80,40 0,80"
        fill="aqua" stroke="red" />
</svg>
```

6.4. Мураккаб шакллар чизиш

Мураккаб шаклларни ягона тег ёки бир неча теглар мажмуаси ёрдамида акслантириш мумкин. Дастрраб битта мукаммал тегнинг имконияти билан танишиб чиқамиз.

6.4.1. PATH теги

Ушбу **path** теги – кесмалар, ёйлар ва әгри чизиклар ёрдамида ҳосил қилинадиган мураккаб шакл чизишга имкон беради. Унинг **d** аломати сатр қийматига эга бўлиб, мураккаб келишувлар асосида шакл чизиш қонуниятини аниқлайди. Қандай қонуниятлар асосида чизикни бошқариб шакл ҳосил қилишни батафсил кўриб чиқамиз.

Буни рассомнинг мўйқаламини юритишга қиёслаш мумкин. Қиймат сифатида **d** аломатига бериладиган сатрда қўйидаги буйруқлар қатнашиши мумкин:

▪ **moveto** — бошлангич нуқта. Қисқача қилиб «M» ёки «m» ҳарфи билан белгилаш мумкин. Таббийки, кўрсатилган ҳарфдан кейин вергул билан ажратилган ҳолда иккита сон: x ва у бўйича координата ёзилади. Бу мўйқаламни даставвал қайси нуқтага текказиш кераклигини кўрсатади.

▪ **lineto** — жорий ўриндан қайси нуқтагача кесма чизиш кераклигини кўрсатади. Синтаксиси «**moveto**»ники сингари.

▪ **horizontal** — жорий нуқтадан ётиқ (уфқий) чизик чизиш учун **H** ёки **h** буйруқлари ишлатилади. Ҳарфдан сўнг битта x бўйича координата кўрсатилади.

▪ **vertical** — жорий нуқтадан тик чизик чизиш учун **V** ёки **v** буйруқлари қўлланилади. Ҳарфдан сўнг битта у бўйича координата кўрсатилади.

▪ **curveto** — жорий нуқта ва **C** ёки с ҳарфларидан кейин кўрсатилган учта координата бўйича ҳосил қилинган тўртбурчак шаклидаги синиқ чизик асосида чизиладиган «Бейзиер» ёйи*.

▪ **smooth** — бу ҳам «**curveto**» сингари ёй чизади, фақат унда **S** ёки **s** ҳарфидан кейин иккита нуқта берилади, учинчиси эса биринчи нуқтанинг иккинчисига нисбатан симметрик акси сифатида қаралади.

▪ **quadratic** — жорий нуқта ва **Q** ёки **q** ҳарфларидан кейин кўрсатилган икки координатадан ўтадиган бурчак асосида чизиладиган «Бейзиер» ёйи.

▪ **smooth quadratic** — жорий нуқта ва **T** ёки **t** дан кейин ёзиладиган битта координата асосида ҳосил қилинган квадрат бўйича чизиладиган «Бейзиер» ёйи.

▪ **elliptic** — эллипс ёйи, бунда **A** ёки **a** ҳарфидан кейин x ва у ўқи бўйича радиуслар, кейин ёйнинг тугаш нуқтаси берилади.

▪ **closepath** — ёпиш, **Z** ёки **z** ҳарфи билан буйруқ берилади, бунда жорий (охирги) нуқта кесма ёрдамида бошлангич нуқтага бириткирилади.

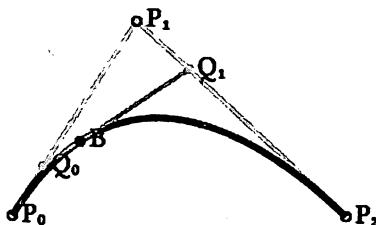
Санаб ўтилган қонуниятларда эътибор қаратиш кепрак бўлган жиҳати, буйруқлар битта ҳарф орқали бе-

рилади. Агар баш ҳарф ишлатилса, тасвир чизиш учун ажратилған умумий майдон бўйича Декарт координаталар тизимидағи x ва у ўқи бўйича мутлоқ координата кўрсатилади. Кичик ҳарф ишлатилса, мазкур (олдинги) нуқтага нисбатан x ва у ўқи бўйича қанча масофага сурилиши лозимлигини англаатади. Соnlар бир-биридан бўш жой ёки вергул билан ажратилади. Агар сон манфий бўлса, минусдан бошқа ажратувчи ишлатиш шарт эмас. Шундай экан, умуман бўш жой ва вергул ишлатмасдан координаталарни фақат «+» ва «-»лар ёрдамида ҳам ифодалаш мумкин.

Кимки, «Бейзиер» ёйи қандайлиги ҳақида тасавурга эга бўлмаса, геометриядан шу мавзуни танишиб олиш тавсия этилади. У мураккаб формула ёрдамида аникланади, қанча нуқталар кўп кўрсатилса, улар ёрдамида ҳосил қилинадиган синиқ чизиққа қараб ёй чизиш шунчак мушкуллашади. 1962 йил Пиере Бейзиер уни оммага маълум қилган бўлса-да, ундан бир неча йиллар илгари Кастельжо томонидан ҳам автомобил кузовлари учун ушбу усул қўлланилган.

Шу ўринда энг содда, фақатгина учта нуқтадан иборат синиқ чизиқ (бурчак) учун «Бейзиер» ёйи қандай чизилиши ҳақида тушунча бериб ўтишни лозим топдик. Бунда бурчакнинг ҳар икки кесмасини тенг (айтайлик н та) бўлакка бўлинади. Кейин шундай вақтинчалик кесма чизилади-ки, унинг бошланғич учи биринчи кесманинг бўлакларга ажратувчи очиқ учига нисбатан биринчи нуқтаси бўлади, иккинчи учи эса бурчакни ҳосил қилган иккинчи кесманинг учига нисбатан биринчи нуқтаси бўлади. Вақтинчалик кесмани ҳам н та бўлакка бўлиб, биринчи нуқтасини маҳсус белгилаб қўйилади. Кейин вақтинчалик кесма ўчирилиб, янгиси ҳудди шу тариқа фақат иккинчи нуқталар асосида чизилади. Шу тариқа вақтинчалик кесмалар қолдирган н та маҳсус нуқталар битта ёйни ҳосил қиласи. Албатта бунда н етарлича катта натурал сон бўлиши керак.

Расм 1. Бейзиер ёйини ҳосил қилиш



Демак, **path** тегининг **d** аломатида кесма ва ёйлар чи-зиш орқали ихтиёрий икки ўйчамли шакл ҳосил қила оламиз. У шунчалик мукаммал экан, бутун бошли тас-вирни фақат битта **path** теги билан чизилса бўладими деган савол туғилиши мумкин. Бўлади, агар шакл фақат битта рангдан иборат бўлиши қаноатлантируса. Одатда бир неча ранглардан фойдаланишга тўғри келади, бу-нинг учун бир нечта тег ишлатилади ва ранглар бўлим бошида ифода этилгандагидек [6.3.] кўлланилади.

Кўйида бир жуфт гилос қандай чизилишининг коди-ни келтириб ўтамиш:

```
<svg width="215px" height="274px" viewBox="0 0 215 274">
    <path fill="none" stroke="#7AA20D" stroke-width="6"
        stroke-linecap="round" stroke-linejoin="round"
        d="M55,174c0,0,77-73,82-104 c4-30,46,91,27,121" />
    <path fill="#7AA20D" stroke="#7AA20D"
        stroke-width="4" stroke-linecap="round"
        stroke-linejoin="round" d="M134,62c-1-6,0-12,5-19
        c11-14,23-17,40-17c0,0,22-0,26-20c-2,12-5,24-9,36
        c-6,19-22,48-47,38 C141,75,136,69,134,62z" />
    <path fill="#ED2246" stroke="#ED0046"
        stroke-width="2" d="M164,193 c1-0,3-0, 5-1c8-2,21-
        6,28,3c15,21,5,55-18,66 c-27,12-58-16-56-45
        c1-13, 8-35,26-27 C155,191,159,193,164,193z" />
    <path fill="#ED2222" stroke="#CB1111"
        stroke-width="2" d="M55,176 c1+0+3+0+5-0c8,
        0,22-2,26,9c9,24-6,55-32,60 c-29,5-53-29-44-56c4-13,
        15-33, 31-20C47,173,50,176,55,176z" />
</svg>
```

6.4.2. Маркердан фойдаланиш

Кўпинча айни бир тасвирни такрорий чизмаслик учун уни маълум ном билан сақлаб қўйиш лозим бўлади. Кейин ундан бир неча бор фойдаланиш мумкин. Бунинг учун **marker** тегини ишлатилади. У қўйидаги аломатларга эга:

- **viewBox** — маркерни суриш ёки масштабини ўзгартириш [6.2.]
 - **markerWidth** — маркер майдонининг эни
 - **markerHeight** — маркер майдонининг бўйи
 - **refX** — маркерни x ўқи бўйича суриш (мусбат сон берилса чапга)
 - **refY** — маркерни y ўқи бўйича суриш (мусбат сон берилса тепага)
 - **markerUnits** — чегаравий чизик қалинлигига қараб маркернинг масштабини аниқлайди. Одатий қиймати «stroke-width»да берилгани каби.
 - **orient** — уфққа нисбатан маркерни буриш бурчаги. Қиймати сон ёки «auto» (90°)

```
<svg>
    <marker id="mCircle" refX="4" refY="4"
        markerWidth="8" markerHeight="8" >
        <circle cx="4" cy="4" r="2"
            style="stroke: none; fill:#000000;" />
    </marker>
    <marker id="mArrow" refX="4" refY="2"
        markerWidth="8" markerHeight="8">
        <path fill="#000000" d="M2+2L6+2L4+6z" />
    </marker>
    <path d="M10+20L40+20S90+20,90+90" stroke="#6666ff"
        stroke-width="4" fill="none"
        marker-start="url(#mCircle)"
        marker-end="url(#mArrow)" />
</svg>
```

Битта **marker** ичида бир неча теглар ёрдамида маҳсус шакл ҳосил қилиш мумкин. Қайси шаклга маркер улаш лозим бўлса, унга мос равищда **marker-start**, **marker-end** ёки **marker-id** аломатлари ёрдамида аввал эълон

қилинган номдагиси бириктирилди. Ушбу аломатлардан бири ёрдамида маркер чакырилмаса, ундаги тасвир акслантирилмайди. Маркернинг ичидағи теглар ягона тасвир сифатида гурухланган ҳисобланади.

Умуман олганда мұраккаб тасвирларни бир неча теглар ёрдамида чизиб, *g* теги ёрдамида гурухлаш мүмкін.

6.4.3. Тасвирни ўзгача шакллантириш

Хар бир тасвирни ёки уларнинг гурухини шакллантиришда маълум қоида бўйича ўзгаришини таъминлаш мүмкін. Масалан: уларни буриш, ҳажмини ўзгартириш, тескари ёки қарама-қарши акслантириш. Ушбу мақсадда барча шакл ҳосил қилувчи ёки уларни гурухловчи тегларда *transform* аломатини қўлласа бўлади. У 3x3 матрица ёрдамида координаталарни қайта ишлаб ҳосил қиласди. Қуйидаги қийматларни қабул қиласди:

- **matrix(<a> <c> <d> <e> <f>)** – шу олтита сонга яна 0, 0, 1 ни кўшиб 3x3 матрица ҳосил қиласди ва унга координатани матрица сифатида кўпайтириб, янги координаталар олади.
- **translate(<x> [<y>])** – x ва у ўқлари бўйича суриш. Қиймат берилмаса 0 деб олади.
- **scale(<x> [<y>])** – x ва у ўқлари бўйича масштаблаш. Агар у берилмаса, хнинг қийматига teng деб ҳисобланади.
- **rotate(<a> [<x> <y>])** – кўрсатилган координата (берилмаса (0, 0) олинади) марказ сифатида а градусга бурилди.
- **skewX(<a>)** – x ўқи бўйича а градусга оғдириш.
- **skewY(<a>)** – y ўқи бўйича а градусга оғдириш.

Мисол:

```
<svg width="180" height="200">
    <text x="10" y="60"
        transform="translate(36) rotate(45 30 30)">
        Бурилган матн
    </text>
</svg>
```

6.4.4. Туслантириш

Шакл рангини туслантириш учун `linearGradient` ёки `radialGradient` тегларидан фойдаланилади. Биринчисида түгри чизик бүйича биттә рангдан иккінчисига аста ўтиб борса, иккінчиси марказдан доира тарзда рангнинг ўзгариб боришини таъминлайди. Улар күйидаги аломатларга эга:

- `x1, y1, x2, y2` — мос равищда бошланғич ва охирги нұқталар координаталари, биринчи ранг қаердан бошланиб, иккінчи ранг қаерда тугашини күрсатади. Қиймати: сон ёки фоиз. Уларнинг қиймати ёрдамида чизикли туслагичнинг йұналишини ҳам аниклаш мүмкін. (`radialGradient` тегида `x1,y1` ўрнига `fx,fy; x2,y2` ўрнига `cx, cy` ва радиус `r` ишлатилади,)

- `gradientUnits` — берилған координаталарни қандай қабул қилинишини күрсатади. Аниқланған турдаги қиймат қабул қилади:

- `userSpaceOnUse` — координаталар (`x1,y1,x2,y2`) умумий SVG обьектта нисбатан

- `objectBoundingBox` — одатий қиймат, координаталар айни тегишли шаклга нисбатан

- `spreadMethod` — тусланувчи ранглар майдони шаклнинг ҳажмидан кичик бўлган тақдирда, бўш қолиши лозим бўлган қисм қандай акслантирилиши лозимлигини күрсатади. Қиймати аниқланған тур:

- `pad` — одатий қиймат. Бошланғич ва охирги ранглар бўш майдонга ёйилади.

- `repeat` — тусланаётган ранглар такрорланади.

- `reflect` — тусланаётган ранглар кўзгудаги акс сингари такрорланади.

- `gradientTransform` — тусланувчи ранглар шаклини ўзгартириш. [6.4.3.]

Айнан ранглар эмас, қандайдир шакллар ҳосил қилиниб орқатасвир намоён этилиши лозим бўлса, унда `pattern` тегидан фойдаланилади. У ҳам юқорида танишиб ўтилган стандарт аломатларга эга: `x, y, width, height` ва `patternUnits` (`gradientUnits` сингари).

Тусланишни тасвиrlашга умумий мисол:

```

<svg width="400" height="120" viewBox="0 0 400 120">
    <radialGradient id="cGradient">
        <stop offset="10%" stop-color="gold" />
        <stop offset="95%" stop-color="green" />
    </radialGradient>
    <linearGradient id="rGradient" x1="0%" y1="0%"
                    x2="100%" y2="100%">
        <stop offset="0%" stop-color="aqua"
              stop-opacity="1" />
        <stop offset="100%" stop-color="blue"
              stop-opacity="1" />
    </linearGradient>
    <pattern id="tus" width="40" height="40"
              x="9" y="9" patternUnits="userSpaceOnUse">
        <circle cx="20" cy="20" r="20" fill="#BBC42A" />
    </pattern>
    <circle fill="url(#cGradient)" cx="60" cy="60" r="50"/>
    <rect fill="url(#rGradient)" x="120" y="10"
          width="100" height="100" />
    <rect x="230" y="10" width="100" height="100"
          stroke="#333333" fill="url(#tus)" />
</svg>

```

6.5. Ҳаракатланувчи тасвиirlар

Шаклларни ҳаракатлантириш учун бир неча махсус теглардан фойдаланилади.

6.5.1. SET теги

Энг оддий ҳаракатни таъминлаш учун **set** тегидан фойдаланилади. У белгиланган вақтдан сўнг шаклнинг кўрсатилиган аломат бўйича қийматини бошлангич ҳолатидан ўзгартиради. Яъни маълум вақтдан сўнг шакл бир кўринишдан иккинчисига дарҳол ўтиб қолади. Ушбу тег қуидаги аломатларга эга.

- **attributeName** — қайси аломат бўйича ўзгаришини кўрсатади. Қиймати: сатр.

▪ **attributeType** — күрсатилган аломат қандай аниқланганини билдиради. Қиймати: аниқланган тур, агар аломат сифатида күрсатилган бўлса – «XML»; шамойили фақат CSS ёрдамида аниқланган бўлса – «CSS» қиймат берилиши керак.

▪ **to** — «attributeName»да күрсатилган аломатнинг қиймати неччига ўзгаришини аниқлайди. Қиймати унда күрсатилган аломатники сингари бўлади.

▪ **begin** — қанча вақтдан сўнг ўзгаришини аниқлайди. Қиймати: вақт.

Бир сониядан кейин тўрг баробар катталашиб қолувчи доирага мисол:

```
<svg width="160" height="160">
  <circle cx="80" cy="80" r="20" stroke="none" fill="#0000ff">
    <set attributeName="r" attributeType="XML"
      to="80" begin="1s" />
  </circle>
</svg>
```

6.5.2. ANIMATE теги

Такорий ҳаракатни таъминлаш учун **animate** теги ишлатилади. Унда юқорида [6.5.1.] изоҳланган аломатлардан ташқари қўйидаги қўшимчаларга ҳам эга:

▪ **from** — қайси қийматдан ҳаракат бошланишини кўрсатади.

▪ **dur** — ҳаракат қанча вақт давом этиши кераклигини кўрсатади.

▪ **repeatCount** — қайтарилиш сони. Қиймати бутун сон ёки «indefinite» (тўхтовсиз).

Уфқий сурилувчи доира учун мисол:

```
<svg width="560" height="60">
  <circle cx="30" cy="30" r="25" stroke="none" fill="#0000ff">
    <animate attributeName="cx" attributeType="XML" dur="2s"
      begin="0s" from="10" to="520" repeatCount="indefinite" />
  </circle>
</svg>
```

Күп браузерлар вақт ўтган сайин рангни ўзгартириб туришни таъминловчи «animateColor» тегини қўлламайди. Шунинг учун, «animate» тегининг ўзидан фойдаланган ҳолда, «attributeName»га ранг қабул қилувчи аломатнинг номини бериб қўллаш тавсия этилади.

6.5.3. AnimateMotion теги

Шаклни маълум йўналишда ҳаракатлантириш учун **animateMotion** тегидан фойдаланилади. Бунинг учун **path** тегидаги қонуниятлар асосида маҳсус чизик чизилади ва шакл унинг бошидан охиригача кўрсатилган вақтда ҳаракатланишини таъминлайди. Ушбу тегда ҳам «animate»ники сингари аломатлар мавжуд. Фақат қўшимча «**path**» аломатига эга бўлиб, у орқали маҳсус қоидалар [6.4.1.] асосида эгри чизик берилади. Мисол:

```
<svg width="320" height="120">
  <path d="M10,50q60,50+100,0q60,-50+100,0"
        fill="none" stroke="green"/>
  <rect x="0" y="0" stroke="#ff0000"
        width="30" height="15" fill="none">
    <animateMotion begin="0s" dur="10s" rotate="auto"
      path="M10,50q60,50+100,0q60,-50+100,0"
      repeatCount="indefinite" rotate="auto" />
  </rect>
</svg>
```

Бунда **path** теги фақат қандай из (траектория) бўйича ҳаракат бўлаётганини браузерда кўрсатиш учун ёзилган холос. Аслида уни ёзмаса ҳам бўлади. Исталган дараҷадаги мураккаб чизик бўйлаб кўрсатилган вақтда ихтиёрий шаклни ҳаракатлантириш мумкин, **rotate="auto"** қиймати ҳаракатланиш пайтида мос равища бурилишни ҳам таъминлайди. Умуман олганда, ушбу аломатта бутун сон градус хисобида берилса ҳам бўлади.

Агар **path** теги ифодалаган из кўриниши шарт бўлса, уни аниқловчи сатр юқоридагидек икки марта такрор ёзилиши мукаммал дастурлаш талабларига жавоб бермайди. Аслида «из»ни бир марта аниқлаб, керак бўлса унга бошқа

ерда яна мурожаат қилиш мумкин бўлиши керак. Бунинг учун **mpath** тегидан фойдаланилади. Мисол:

```
<svg width="320" height="120">
  <path id="yol" d="M10,50q60,50+100,0q60,-50+100,0"
        fill="none" stroke="green"/>
  <rect x="0" y="0" width="30" height="15"
        stroke="#ff0000" fill="none">
    <animateMotion begin="0s" dur="10s"
                    repeatCount="10" rotate="auto">
        <mpath href="#yol"/>
    </animateMotion>
  </rect>
</svg>
```

6.5.4. AnimateTransform теги

Шаклни **transform** аломати [6.4.3.] бўйича ҳаракатлантиришда **AnimateTransform** теги қўлланилади. У «attributeName», «attributeType», «from», «to», «begin», «dur», «repeatCount» каби стандарт аломатларга қўшимча «type»га ҳам эга. Бу «type» аломатида қандай ўзгариш содир бўлиши кўрсатилади. Қиймати – аниқланган тур: matrix, translate, scale, rotate, skewX, skewY. [6.4.3.]

Унинг имконияти ёрдамида шаклни ҳаракат ёрдамида суриш, ҳажмини ўзгаририш, оғдириш, буриш ва маҳсус кўринишида акслантириш мумкин. Мисол:

```
<svg width="200" height="200" style="border:1px solid red;">
  <rect x="30" y="30" width="60" height="40"
        fill="none" stroke="green">
    <animateTransform type="rotate"
                      attributeName="transform" dur="3s"
                      from="0 100 100" to="360 100 100"
                      begin="0s" repeatCount="indefinite" />
  </rect>
</svg>
```

Кейинги «JavaScript»га бағишлиңган китобимизда нағақат шаклларни, балки барча обьектларни ҳаракатлантириш ва айлантириш, ранг ва қўринишларини ўзгартириш, яшириш ва пайдо қилиш, ўчириш ва яратиш ҳақида батафсил ёритиб ўтилади.

Умуман олганда SVG жуда бой имкониятларга эга алоҳида тил сифатида шаклланган. Ҳозирча унинг 1.1 талқинидаги кўп имкониятлари браузерларда қўлланилади. Биз уларнинг асосини ташкил қилувчи мухим теглар билан танишдик, улар орқали керакли шаклни ҳосил қилиш ва ҳаракатлантириш мумкин. Ушбу китобда ёритилмаган SVGнинг қолган теглари HTML ва CSSнинг аломатларига ўхшаб кетади. Шунинг учун ҳар бирига алоҳида тўхталишни лозим топмадик.

VII БҮЛİM

ФОРМА ЭЛЕМЕНТЛАРИ



7.1. Форма яратиш

Маълумот ва қийматларни сақлаш, киритиш, қайта ишлаш, таҳрирлаш, муҳими серверга юбориш учун формадан фойдаланилади. Форманинг ичида ёзиладиган теглар асосида акслантириладиган объектларни – элементлари, серверга қиймат юбориш учун хизмат қиласидиган объектларни эса – дарчалари деб юритамиз. Форма **form** теги ёрдамида яратилиб, ўз ичига маълум қийматларни қабул қиласидиган дарчаларни мужассамлаштирган кути вазифасини бажаради. Серверга маълумотлар шу кути хусусиятига қараб жамланиб юорилади. Ушбу тег қуийдаги аломатларга эга:

- **accept-charset** – юборилаётган ёзувлар кодламасининг номи, қиймати: satr.
- **action** – қийматлар юбориладиган сервер манзили, қиймати: ишпорат. Кўрсатилмаса, жорий веб-саҳифа манзили олинади.
- **autocomplete** – фойдаланувчи дарчаларга олдин қиймат киритган бўлса, веб-саҳифа қайта юклангандан кейин даставвал объектларда ўша қийматлар қўйилишини таъминлайди. Мантиқий тур қабул қиласи.
- **enctype** – форма маълумотларни юборишдан аввал қандай кодлашни аниқдайди, қиймати аниқланган тур:
 - **application/x-www-form-urlencoded** – Одатий қиймат, сатрлардаги белгиларни маҳсус келишув асосидаги «**urlencoded**» функцияси ёрдамида кодлайди. Унда бўш жой «+», лотин ҳарфларидан бошқа белгилар эса «%» билан ўн олтилик саноқ тизимидағи маҳсус рақами

орқали ёзилади. Масалан, «Китоб» сўзи:
«%D0%9A%D0%B8%D1%82%D0%BE%D0%B1» тарзида.

- `text/plain` – фақат бўш жой «+»га алмаштирилади, маълумот кодланмайди.
- `multipart/form-data` – маълумот кодланмайди, файл юборишни таъминлайди.

■ `method` – қийматларни серверга юбориш усули. Аниқланган тур:

○ `get` – қийматларни дарчаларнинг номи ортидан «==» белгисидан кейин ёзиб, «&» белгиси билан бирлаштириб, «action»да кўрсатилган ишоратдан сўнг «?» белгиси билан улаб, янги ишорат ҳосил қиласди ва шу йўсинда маълумот юборади. Масалан, «a» номли дарчага «1», «b» номли дарчага «www» қиймат берилган бўлса, юбориладиган манзил ишоратига «?a=1&b=www» сатри қўшиласди. Бунда барча юбориляётган маълумотлар фойдаланувчи томонидан ўқиш ва ўзгартиришга қулай. Албатта жуда узун ва катта ҳажмдаги қийматларни ушбу йўл билан юбориш мақсадга мувофиқ эмас, унда кўпи билан 4 Кб маълумот жўнатиш мумкин.

○ `post` – формадаги дарчаларнинг қийматларни «enctype»да кўрсатилган кодлаш асосида фойдаланувчига кўринмайдиган тарзда юборишни таъминлайди. Албатта, бунда серверда ўрнатилган маҳсус чегарагача бўлган ҳажмдаги қийматларни ва ҳатто файлларни юбориш имконияти мавжуд.

○ `put` – маълумотларни файл матнида юбориш. Умуман олганда форма ёрдамида серверга ҳамма маълумотларни «enctype=multipart/form-data» қиймат бериш орқали алоҳида файл сифатида ёки «get» услубида келтирилгани каби сервер манзилига қўшимча тарзда ёзиш билан ва «post» услубида ёритилган каби алоҳида бириктириш орқали юборишлардан ташқари, йўналтирилаётган файлнинг тавсиф хабари [2.3.] ичида қўшишиб ёки таркибидаги матни орқали бериш ҳам мумкин.

- **name** – форманинг номи. Агар веб-саҳифада бир неча форма иштирок этса, уларнинг номлари тақорорлан-майдиган бўлиши талаб этилади. Скрипtlар ёрдамида формага унинг номи бўйича мурожаат қилишни таъминлайди. Қиймати: идентификатор.

- **noValidate** – одатда дарчаларга киритилган қийматлар юборищдан олдин талаб этилган андозага тўғри келиши текширилади, агар ушбу аломат кўрсатилса, текширув ўчирилади.

- **target** – сервердан қайтган жавоб қаерда аксланишини кўрсатади. [2.4.]

Юқорида маълумот юбориш «method»ини уч хил усули билан танишдик. Аслида улар ўндан ортиқ ва яна қўшилиб борилиши мумкин. Ҳаммасининг юбориш технологияси шу учала услубдан айнан биридир, фақат номланиши бошқача. Сабаби сервер услуг номига қараб унга мос амал бажаради. Аслида ҳеч қандай янгида технология яратмасдан, сервер билан маълумот алмашибни маълум қолипга солиш мақсадида HTTP қайдномаси асосчиларидан бири Рой Томас Филдинг 2000 йилга келиб REST (Representational State Transfer – тақдимот ҳолатини юбориш) атамасини киритди. Унга кўра юбориш услубларининг номи кенгайди: post, get, put, delete, patch, view, copy, head, link, unlink, purge, lock, unlock, propfind, options. REST услубияти билан тўлақонли ишлаш кейинги китобларда батафсил ёритилади.

7.2. INPUT дарчаси

Форма элементлари ичида энг мураккаби **input** дарчаси ҳисобланади, у кўрсатилган турига қараб бир неча хил қийматларни қабул қиласи ва обьектнинг кўрининши ҳам шунга мос равища ўзгаради. У ёпилиш қисм мавжуд бўлмаган **input** теги орқали ифодаланади, яъни барча хусусиятлари фақат аломатлар ёрдамида берилади.

Даставвал, унинг барча – яъни HTML5да янги қўшилган ҳамда эскирган деб ҳисобланмаган, аломатлари билан танишиб ўтамиш:

▪ **accept** — фақат файл юборилаёттан ҳолда ушбу аломатда МІМЕ тури [2.4.] ёки унинг андозаси кўрсатилади. Масалан: товуш файллар учун «audio/*», расм файллар учун «image/*» сифатида кўрсатиш мумкин. Бунда танланган файл кўрсатилган андозага мос келмаса, у танловлар ойнасида намоён бўлмайди. Қиймати сатр, катта-кичик ҳарфлар аҳамиятга эга эмас. Бир неча МІМЕ тур кўрсатилиши лозим бўлса, вергул билан ажратилади.

▪ **alt** — агар `input` тури расм сифатида кўрсатилган бўлса, расм акслантирилмайдиган ҳолларда ушбу аломатда ёзилган матн кўринади. Қиймати: сатр.

▪ **autocomplete** — ўзи тўлдириш имкониятини ёқиши ёки ўчириш. [7.1.]

▪ **autofocus** — дарчани фаоллаштиради (бўш жой ёки киритиш тутмаси босилганда ишлашга шай ҳолда туради). Ушбу аломат битта веб-саҳифадаги элементларнинг фақат биттасита берилиши тавсия этилади. Чунки доим курсор (хатчўп) босилган, яъни айни пайтдаги фаол дарча ягона бўлади. Акс ҳолда ушбу аломатли дарчаларнинг энг охиргиси фаол бўлади. Зарурмас қиймат қабул қиласади.

▪ **checked** — `input` тури танланувчи ёки белгиланувчи сифатида кўрсатилган бўлса, яратилаётганда рост қийматни қабул қилишини англатади (зарурмас қиймат).

▪ **dirname** — қиймати: сатр. Серверга унда кўрсатилган сатр номи билан «ltr» ёки «rtl» қийматларидан [2.7.] бири юборилади. Ушбу аломатни Internet Explorer ва Firefoxлар қўлламайди. Айрим дарчаларда араб ёзуви ишлатилиши мумкин бўлган ҳоллардагина бу имконият ас қотиши эҳтимоли бор ва у ҳам жуда кам.

▪ **disabled** — дарчанинг фаоллигини ўчиради. Маълумот киритиш ёки ўзгартириш имконияти бўлмайди (зарурмас қиймат).

▪ **form** — ушбу аломатда форманинг «id»сидаги қиймат берилади. Агар дарча «form» тегининг ичидаги жойлашган бўлса, бу шарт эмас. Баъзи ҳолларда элемент формадан ташқарида (бошқа файлда) яратилган бўлиши мумкин, шунда ташқаридағи дарчани формага бириктириш имкониятини яратади. Лекин баъзи браузерлар буни қўлламайди ва бундай ишлатиш тавсия этилмайди.

▪ **formaction** — ушбу аломат формадаги «action» билан бир хил вазифани бажаради. Битта формада бир неча тутма қўйилиб, ҳар бирини босганда ўзига хос бошқа-бошқа манзилга мурожаат қилиш лозим бўлса, унда шу аломатдан фойдаланилади ва у устивор ҳисобланади. Яъни **input** тури жўнатишга мўлжалланган тутма бўлса; «**formAction**», «**formEnctype**», «**formMethod**», «**formNoValidate**», «**formTarget**»лар мос равишда форманинг аломатларига муқобил бўлади.

▪ **list** — «**datalist**» тегининг «**id**»си. [7.4.4.] Маълумот киритилаётганда рўйхатдан мос сатрни танлаш имкониятини беради.

▪ **max** — сон ёки сана сифатидаги киритилиши мумкин бўлган энг катта қийматни аниқлайди.

▪ **maxlength** — киритилаётган сатрдаги белгиларнинг энг кўп миқдори нечта бўлишини чегаралайди, қиймати: сон.

▪ **min** — сон ёки сана сифатидаги киритилиши мумкин бўлган энг кичик қийматни аниқлайди.

▪ **minlength** — киритилаётган сатрдаги белгилар энг камида нечта бўлиши (сон).

▪ **multiple** — қийматлар бир неча бўлиши мумкинлигини кўрсатади. Бунда улар вергул билан ажратилиши керак. Синтаксис текширилаётганда бу муҳим.

▪ **name** — дарчанинг номи. Битта формадаги дарчалар номи такрорланмаслиги шарт. Ушбу ном билан унинг қиймати серверга юборилади. Скрипtlар ҳам шу ном билан мурожаат қилиши мумкин. Қиймати: идентификатор.

▪ **pattern** — муентазам ифодалар ёрдамида киритила-диган қийматга андоза бериш. Қиймати: муентазам ифода қоидалари бўйича (бу ҳақида дастурлашга бағищланган китобимизда батафсил танишамиз) ёзилган сатр. Мисол тариқасида фақат бешта бош лотин ҳарфлари ёзиладиган дарчани қарасак, у қўйидагича бўлади:

```
<input pattern="^[A-Z]{5}$">
```

▪ **placeholder** — қиймати: сатр. Ушбу аломатдаги матн хирароқ ранг билан бўш дарчага ёзилиб туради. Бирор қиймат берилмаганда, нимани киритиш лозимлиги

ҳақида хабар беріб туриши учун құлланыладиган жуда қулай восита.

HTML	<code><input placeholder="Исмиңгизни кирилинг"></code>	Натыжа	Исмиңгизни кирилинг
------	--	--------	---------------------

- **readonly** — қийматни ўзgartириш имконидан маҳрум этади (зарурмас қиймат).
- **required** — дарчага маълумот киритишни мажбурий килиб қўяди. (зарурмас қиймат). Акс ҳолда форма серверга юбориш учун мурожаат қилмайди.
- **size** — дарчанинг кенглиги. Қиймати: сон, нечта белги сифишини кўрсатади.
- **src** — input тури тасвир сифатида аниқланган бўлса, ушбу аломатда расмнинг манзили кўрсатилади.
- **step** — input тури сон сифатида кўрсатилган бўлса, ундаги қийматни пайдо бўладиган қўшимча милларни босиш орқали ошириш ёки камайтириш қадами.
- **value** — дарча яратилаётганда қандай қиймат билан аксланишини кўрсатади.
- **type** — дарча қандай турдалигини аниқлайди. (Куида батафсил танишамиз.)

7.3. INPUT дарчаси турлари

INPUTнинг аломатлари билан танишиб чиқилганда маълум бўлди-ки, уларнинг кўпчилиги фақаттинга кўрсатилган турдаги дарчаларгагина алоқадор. Масалан, матн ёзиладиган дарча учун «step» ҳеч қандай вазифани бажармайди, вақтни аниқлайдиган дарчага «checked» қўйиш маъносиз. Бундай ортиқча аломатлар ёзилган ҳолда, HTMLнинг қоидасига мувофиқ, шунчаки эътиборсиз қолдирилади.

Даставвал HTML4гача мавжуд бўлган турлар билан танишиб ўтамиз, сабаби HTML5да қўшилган кўпгина фойдали турдаги дарчалар баъзи браузерларда ҳанузгача қўлланмайди. Хуласа, inputнинг type аломати қуийдаги қийматларни қабул қиласи:

- **button** — тутма ҳосил қиласи.
- **checkbox** — белгиланувчи (бир нечта белгилаш мүмкін) дарча ҳосил қиласи.
- **file** — файл таңлаб юборишни таъминлайды.
- **hidden** — яширин дарча, фойдаланувчига күрингідейді.
- **image** — расмдан иборат дарча, уни босилғанда форма серверга жүнатылади.
- **password** — ҳар бир белгиси юлдузчадан иборат яширин сатр.
- **radio** — таңланилувчи дарча ҳосил қиласи. Одатта бир неча ёзувлардан фақат биттаси таңланади, шуннинг учун ушбу турдаги дарчаларнинг бир нечаси кетма-кет келади ва бир хил ном (*name*) билан ифодаланылади.
- **reset** — формадаги барча қийматларни бошланғич ҳолатига келтирувчи тутма.
- **submit** — форма (маълумотлари)ни серверга юборувчи тутма.
- **text** — бир қатор матн ёзишга мүлжалланған дарча.

Энди HTML5да қўшилган янги турлар билан танишамиз, шу билан бирга оммабоп браузерларнинг қайси талқинларидан бошлаб қўлланишини ҳам кўриб ўтамиз. Қайси браузерларда ҳалигача тегишли турдаги дарчани акслантирувчи имконият мавжуд бўлмаса, унга «—» белгиси кўйилган ва яқин келажакда бу камчиликлар бартараф этилишига умид қилиб қоламиз.

Киймат	 IE	 Edge	 Chrome	 Opera	 Safari	 Firefox	 Android	 iOS
color	—	14	20	11.5	—	29	4.4	—
date	—	13	20	10.1	—	—	4.4	5.0
datetime	—	11	20	10.1	—	—	4.4	5.0
datetime-local	—	11	20	10.1	—	—	4.4	5.0
email	10	11	5	10.1	5	4	3	3.1
month	—	13	20	10.1	—	—	4.4	5.0
number	10	11	6	10.1	5	29	4	4.0
range	10	11	4	10.1	3.1	23	4.3	5.0

Қиймат	 IE	 Edge	 Chrome	 Opera	 Safari	 Firefox	 Android	 iOS
search	10	11	5	10.1	5	4	2.3	4.0
tel	10	11	5	10.1	5	4	2.3	3.1
time	—	13	20	10.1	—	—	4.4	5.0
url	10	11	8	15	6	4	4	3.1
week	—	13	20	10.1	—	—	4.4	5.0

Дарча url, tel ёки email турида бўлса, киритилган сатр ўзимиз билган мос қолипга тушишини браузер формани юбормасдан олдин текширади. Агар search турида бўлса, унинг text туридан фарқи — киритилаётган белгилар ёзилганда «x» пайдо бўлиб, уни босганда матнни ўчириб юбориш имконини ҳам яратади. Қолган барча турдагилар учун браузер кўрсатилган номга мос ва ўзига хос андозада ёзилишини таъминловчи, танлаш имконияти билан киритиладиган қийматлар ҳосил қиласди.

Муаммо шундаки, вақтни кўрсатадиган турдагилар учун сана шаклини қай тарзда аксланиши лозимлигини кўрсатиш мумкин эмас. Масалан, америкаликлар санани ифодалашда аввал ой рақамини кейин кун рақамини ёзишади, биз кунни биринчи ёзишга одатланганмиз. Шунинг учун ҳам энг оммабоп фойдаланиладиган «Mozilla Firefox» ҳалигача улар учун интерфейс қилмагандир. Умуман олганда, қўшимча кутубхоналар ёрдамида (масалан, JQueryUI) ушбу муаммога ечим топиш мумкин.

7.4. Форма дарчалари

Энди форманинг inputдан бошқа теглар ёрдамида яратиладиган дарчалари билан танишамиз. Уларнинг ҳаммаси учун юқорида келтириб ўтилган аломатлардан «autofocus», «autocomplete», «name», «form», «formaction», «formenctype», «formmethod», «readonly», «formnovalidate», «formtarget», «required», «disabled» ва «value»лар тавсифлангандай вазифани бажаради. Куйида булар ва универсал аломатлардан ташқари, фақат ўзига хослари келтирилиб ўтилади.

7.4.1. Тутма

Тутмани юқорида `<input type="button | reset | submit">` каби кўринишда яратилиш мумкинлигини кўриб ўтдик. Лекин ундан ташқари алоҳида `button` теги ҳам мавжуд бўлиб, ўзининг ичида бошқа теглар ёзишга рухсат этади. Бу — тутманинг ичида расм қўйиш, ёзувининг шакли ва рангини ўзгартириш, қолаверса, бошқа керакли объектларни акслантиришга имкон беради. У қўйидаги ўзига хос аломатга эга:

- `type` — тутманинг вазифасини англатади. Қиймати аниқланган тур:
 - `button` — оддий тутма.
 - `reset` — формадаги қийматларни бошланғич ҳолатига келтиради.
 - `submit` — форма маълумотларини серверга юборади (оддатий қиймат).
 - `menu` — `menu` теги [3.4.2.] билан яратилган рўйхат «`id`»си, тутма босилганда танлаш учун бириткирилган меню намоён бўлади.

Мисол:

```
<button>
    
        Суратли тутма
</button>
```

7.4.2. Матн ёзиш дарчаси

Бир неча қатордан иборат матнлар ёзиш учун `textarea` тегидан фойдаланилади. У қўйидаги ўзига хос аломатларга эга:

- `cols` — эни (кенглиги), қиймати: сон, белги сифиши хисобида.
- `rows` — бўйи (баландлиги), қиймати: сон, белги сифиши хисобида.
- `maxlength` — матндаги белгиларнинг кўпи билан нечта бўлиши, қиймати: сон.
- `wrap` — кейинги сатрга қай тарзда тушишни созлайди. Қиймати: аниқланган тур:
 - `soft` — оддатий қиймат. Бунда сатрга сифмаган матн-

нинг давоми кейинги қаторга ўз-ўзидан тушади ва серверга юборилаётганда ушбу қаторлар бўлинишини махсус белги сифатида жўнатилимайди, фақат киритиш тутмаси (enter) ёрдамида қаторлар бўлинган бўлса, улар юборилади.

◦ **hard** – Бу қийматни кўрсатишдан аввал «rows» аломатига сон берилиши талаб этилади. Бунда ҳар бир қатор бўлиниши махсус белги (`n`) билан серверга юборилади. Киритиш тутмаси (enter) атайин босилгани ёки автоматик қўшилдими, фарқлаш имконсиз.

◦ **off** – кейинги қаторга ўтказишни ўчиради. Барча матн битта сатрдан иборат бўлиб, майдонга сифмаган ҳолда айлантиргич (scroll) пайдо бўлади.

7.4.3. Тайинланувчи (select) дарча

Юқорида `input` теги ёрдамида танланувчи (radio) ва белгиланувчи (checkbox) дарчалар қандай қўйилиши билан танишиб ўтдик. Улар келтирилган рўйхатдан бир ёки бир нечасини ажратиб олиш учун фойдаланилади ва унда рўйхатдаги ҳамма элементлари кўриниб туради. Тайинланувчи дарчада одатда фақат рўйхатнинг белгиланган элементинигина кўрсатилади. (Албатта созлаш орқали бир неча қаторни кўрсатадиган килиш мумкин, бунда сифмаган элементлар учун айлантиргич пайдо бўлади.) Барча рўйхатни ёрдамчи «мил» тутмасини босиш орқали кўрилади. Бундай очилувчи рўйхатини тайинланувчи дарча деб номлаймиз ва у `select` теги ёрдамида яратилади. Ўзига хос аломатлари:

- **multiple** – бир нечасини танлашга имкон беради, зарурмас қиймат.

- **size** – нечта элемент (қатор) кўриниши кераклигини аниқлайди, қиймати: сон.

Тайинланувчи дарчадаги рўйхатнинг элементлари махсус `option` теги ёрдамида берилади. У куйидаги аломатларга эга:

- **disabled** – элементни нофаол қиласди.

- **selected** – яратилган пайти қайси элемент танланиб турғини кўрсатади.

- **value** – элементнинг кўринмайдиган асл қиймати.

У берилмаган ҳолда тег ичидаги ёзувни қабул қиласи. Рўйхатдаги ёзувлар узун бўлиши мумкин, шунинг учун уларни қиймат сифатида қарашиб маъкул эмас. Дастурлов-чиларга қайси элементларни танланганлигини билиш учун уларнинг қиймати бўйича мурожаат қилиш кулади.

Элементлар кўпайиб кетганда уларни ичидан саралаб олиш мушкуллашади. Алифбо тартибида рўйхатни саралаш ҳам доимо самара беравермайди, сабаби номланишлар турли соҳаларга оид бўлиши мумкин. Бундай вазиятларда элементларни маълум мавзуларга ажратиб гуруҳлаш лозим бўлади. Гуруҳлаш учун маҳсус `optgroup` теги ишлатилади. Унинг нофаоллаштирувчи «disabled» ва гуруҳ номини кўрсатувчи «label» аломатлари мавжуд.

Мисол:

```
<select size="2" multiple>
    <optgroup label="Group 1">
        <option>Option 1.1</option>
    </optgroup>
    <optgroup label="Group 2">
        <option selected>Option 2.1</option>
        <option>Option 2.2</option>
    </optgroup>
    <optgroup label="Group 3" disabled>
        <option>Option 3.1</option>
        <option>Option 3.2</option>
        <option>Option 3.3</option>
    </optgroup>
</select>
```

7.4.4. Танланувчи сатр

Сатрни `input` дарchasига тўлиқ ёзиш кўп ҳолларда малол келади. Агар кўрсатилган дарчага киритилиши мумкин бўлган сатрларнинг аксарият матнлари маълум бўлса, улардан мосини танлаб, фақат истисно тариқасидаги ёзувнингина киритиш анчагина қулийлик яратади. Ушбу мақсадда `datalist` теги кўлланилади, унинг ичидаги `<option>` теги ёрдамида берилади. Мис `input` дарchasига `<datalist>`нинг `<id>`си берилади.

Мисол:

```
Браузер номи: <input list="browsers" name="myBrowser">
<datalist id="browsers">
    <option value="Chrome">
    <option value="Firefox">
    <option value="Internet Explorer">
    <option value="Opera">
    <option value="Safari">
    <option value="Microsoft Edge">
</datalist>
```

7.4.5. Маълумотларни шифрлаш

Кўп ҳолларда, айниқса калитсўзларни ва маҳфий маълумотларни юборишда уларни яширин тарзда жўнатиш долзарб ҳисобланади. Ҳозирда барча браузерлар маҳсус ширфлаш алгоритмларини қўллайди ва keygen теги орқали ифодаланади. Шифрлаш очиқ ва ёпиқ калитлар асосида амалга оширилади, очиқ калит серверга юборилади, ёпиги браузернинг ўзида сақланиб, келаётган ширфланган маълумотларни очища ишлатилиди. Ушбу калитларнинг қийматига қараб айни бир хил маълумот турлича кўринишида кодланган ҳолда келиши мумкин. Одатда шифрлашнинг RSA алгоритми қўлланилади, браузер ва сервер тушунган ҳолда бошқа номдагисини ишлатиш мумкин, ном keytype аломатида кўрсатилади. Форма юборилаётганда қийматлар қай тартибда ўзгаришини эса challenge аломатида кўрсатилади. Мисол:

```
<form>
    <keygen name="name" challenge="__" keytype="rsa" disabled>
</form>
```

Таъкидлаб ўтиш жоизки, ушбу услубдан фойдаланиш яхши самара бермайди. Албатта бу маълумот алмасиши жараёнини секинлаштиради ва хавфсизлик жиҳатдан ҳам кутилган натижа бермайди. Албатта юборилган маълумотни маҳсус теришлар орқали маълум муддатда аниқлаш имконияти мавжуд. Ҳозирда барча маълумотларни ва ҳаттоқи ишоратни ҳам кодлаб юборадиган

«https» қайдномаси кенг қўлланилади. Ушбу технология ишлатилганда рухсат этилган сертификат кўрсатилганда гина маълумотларни очишга имкон яратилади. Ҳозирда бу маълумотлар махфийлигини таъминлашнинг оммабоп, қулай ва ишончли воситасидир.

7.4.6. Жараёнларни кўрсатиш

Веб-саҳифа устида қандайdir амаллар бажарилар экан, баъзан берилган буйруқ қандай натижа билан якунланганини кўриш, агар у узоқ вақт олса, яна қанча кутиш лозимлигини кўрсатиб бориш талаб этилади. Агар кутилган натижа бир сонияда акслантирилмаслиги эҳтимоли мавжуд бўлса, бу ҳақида фойдаланувчини огоҳлантириб, маҳсус кузатиб бориш дарчасини кўрсатиш тавсия этилади. Бунинг учун энг оддий *progress* тегидан ҳам фойдаланиш мумкин. Унинг сон қабул қилувчи иккита аломати бор: «max» ва «value»; «max»та қадамдан иборат аммалларни ҳар бирини бажарилганда «value»га шу қийматни бериб бориш йўли билан ёки ҳар маълум миллисонияда бажарилган иш ҳажмига қараб фоиз ҳисобида сон бориш орқали ушбу обьектни акслантириб бориш мумкин. Умуман олганда дастурларни ўрнатиш ёки буйруқ бажарилишини кутиш жараёнида ҳар бир ўқувчи фоизни кўрсатувчи тасмани кўрган деб ишонамиз. Мисол:

```
<progress value="79" max="100">
```

Кўринишидан шунга ўхшаб кетадиган *meter* теги ҳам мавжуд бўлиб, у доимий кузатилиб борилиши лозим бўлган жараёнлар учун кўлланилади. Ўзига хос жиҳати шундаки, унда юқори ва қутиш оралари кўрсатиб қиймат шу оралиқдан чиқадиган ҳолларда огоҳлантириш тарзида бошқа рангта киради. Қутишни аломатларга эга:

- **value** — жорий қиймати;
- **min** — энг кичик қиймат, одатий қиймати: 0;
- **max** — энг катта қиймат, одатий қиймати: 1;
- **low** — қутиш ҳавфли қиймат;
- **high** — юқори ҳавфли қиймат;

- **optimum** — аслида меъёрий қиймати қандай бўлиши лозимлиги;

Мантиқан олиб қараганда ҳақиқий сон қабул қиласидиган ушбу қийматлар «min» ва «max»лар орасида бўлиши талаб этилади. Масалан, юргизгичдаги суюқлик ҳароратининг ўзгариш жараёнини кузатиб борища; агар 80° дан юқори бўлса буни вахимали ҳолат, агар 30° дан паст бўлса бу ўта суст ҳолат сифатида баҳолаш лозимлиги созланса, қўйидагича тег ёзиш мумкин:

```
<meter value="22" max="100" low="30" high="80">
```

Бунда ҳавфли оралиқдаги қийматлар бошқа рангда акслантирилади.

Аслида input сўзининг эгизаги ҳисобланган output номли тег фақатгина HTML5да киритилган. Бу кири тувчи дарча эмас, аксинча чикувчи натижани кўрсатиш учун мўлжалланган обьект. Унгача ҳодисанинг ўз вақтида натижасини чиқариб бориш учун турли теглардан фойдаланиш бироз нокулайлик келтирар эди. Ушбу тег қай даражада самарали эканлигини кўриш учун форма элементларидаги қийматлар ўзгарганда «output»га чикувчи натижани кузатиб бориш имконини берувчи мисол келтирамиз:

```
<form oninput="result.value = a.value * b.value">
    <input type="range" name="b" value="50" > *
    <input type="number" name="a" value="10" > =
    <output name="result">500</output>
</form>
```

Таъкидлаб ўтиш лозимки, форманинг «oninput» ҳодисасига JavaScript коди ёзилган сатр берилган. Ўнда формадаги обьектларнинг қиймати, уларнинг номлари билан қандай олинниши ва улар устида математик амал бажарилиши кўрсатилган холос.

7.5. Формага объектларни жойлаш

HTML тилидаги формани ҳам ўзимиз одатланган операцион тизимдаги ойнага қиёслашимиз түгри бўлади. Ойна ичидағи тутмалар, ёзув ва киритиш дарчалари фойдаланувчига қулай тартибда жойлаштирилган бўлади. Биз ҳам ўз веб-саҳифамизда бунга катта эътибор қаратишимиз лозим. Махсус жилолар бериш орқали веб-формани ҳам операцион тизимдаги ойна кўринишига келтириш мушкул эмас.

Юқорида танишган дарчаларимиз маҳсус тег ёрдамида браузерда обьект яратилишини таъминлайди холос. Лекин фойдаланувчи маълумот киритиши ёки танлаши учун, дарчага қандай қиймат берилиши лозимлигини билиши керак. Бунинг учун ҳар бир дарчага изоҳ сифатида номланиш беришни таъминловчи **label** теги қўлланилади. Унинг «for» номли аломати мавжуд бўлиб, қайси дарча учун эканлигини кўрсатади. Барча формадаги дарчалар учун **label** ёзиш тавсия этилади. Ўндаги ёзувга босилганда тегишли дарчага босилгандай самара беради. Бу белгиловчи **input** обьектлари учун жуда қулай, чунки обьектнинг ҳажми кичик, ёзув эса анча узун. Мисол:

```
<label>Белгилаш: <input type="checkbox"></label>
```

Формадаги обьектларни гурухлаш учун **fieldset** тегидан фойдаланилади. Унинг ичидағи элементлар түгри тўртбурчак ичida акслантирилади. Мисол:

```
<form>
  <label for="yoshi">
    Еми: <input name="yoshi" type="number" value="34">
  </label><br>
  <fieldset>
    <legend>Жинси</legend>
    <label>
      <input name="jinsi" type="radio"> Эркак
    </label><br>
    <label>
      <input name="jinsi" type="radio"> Аёл
    </label><br>
  </fieldset>
  <p> <input type="submit"> </p>
</form>
```

Бунда чегаравий чизиқнинг төпасига гурух номини кўйиш учун **legend** теги ишлатилган.

CSS билан танишиш жараёнида объектларни бир текисда бежирим кўйиш учун нималар қилиш кераклигиги батафсил танишиб ўтамиз.

Умуман олганда қийматларни сақлаш учун доим форма яратиш шарт эмас. Одатда object ва applet тегларида маълум қийматларни маҳсус номлар билан param теги ёрдамида сақлаш мумкин. Лекин ушбу ўзгарувчиларни серверга юбориш ёки фойдаланувчи томонидан ўзгартириш маҳсус скрипtlар ёрдамида бошқача тарзда амалга оширилади. Муқаммалроқ тасаввур ҳосил қилиш учун param тегни кўллашга мисол кўриб ўтамиз:

```
<object classid="my.class">
  <param name="manzil" value="http://yurt.uz/">
  <param name="rang"   value="magenta">
  <param name="yoshi"  value="35">
</object>
```

CSS ТАРКИБИ



8.1. CSSни боғлаш

CSS (Cascading Style Sheets — саҳифанинг поғонали шамойили) объектларга безак ва жило беришга мўлжалланган. Замонавий дастурлашда HTML тили фақат саҳифада нима чиқишини ва қандай тартибда жойлашишини кўрсатади ҳолос. CSS эса акслантирилаётган объектларнинг қай тарзда кўринишини ва қанақа намоён бўлиши лозимлигини белгилайди. Бундай имконият HTMLда ёзилган коднинг ҳажмини кичрайтиришга имкон яратади, ҳар бир объектнинг қандай тусда аксланишини унинг тегида ёзиш шарт эмас.

Турли хил қурилмалар учун саҳифанинг ўзига мос кўринища намоён бўлишини таъминлади. Компьютер, смартфон, планшет, телевизор кабиларнинг ўлчамига мос равишда саҳифадаги объектларни жойлаштириш мумкин. Бунда нафақат масштабни ўзгартириш, балки ҳар бир қурилмада уларнинг фойдаланувчилари кўниккан кўринища объектларни кўрсатса бўлади.

CSSдан фойдаланиш тезкорликни оширишда ҳам катта хизмат қиласди. Биринчидан HTML саҳифанинг ҳажми кичиклиги учун у тез юкланса, иккинчидан бир қанча веб-саҳифалар учун ягона CSS файлини қўллаб, код ёзилишини қисқартириш мумкин. Учинчидан, CSS алоҳида файл сифатида юкланса, браузерлар уни ўзида кешлаб (хотирасига сақлаб олиб) ҳар сафар уни юклаб олишга ортиқча вақт ҳам, интернет трафигини ҳам сарфлаб ётмайди. Веб-саҳифанинг ҳажми камлиги нафақат вақт ва трафикдан тежашга фойда беради, балки ёзилган

қонуниятлар камлиги учун уларни браузер эслаб қолиб акслантиришда ҳам тезкорликка әришилади.

CSS уч хил күринищда HTMLга боғланади:

1. Саҳифанинг бош (head) қисмида **style** теги ичида эълон қилиш орқали.

2. Алоҳида «css» кенгайтмали файлда эълон қилиб, саҳифанинг бош қисмида унга мурожаат қилиш орқали:

```
<link rel="stylesheet" href="mystyle.css" type="text/css">
```

3. Ҳар бир тегда мавжуд бўлган универсал аломат стилера CSS қонуниятлари бўйича ёзилган сатр сифатида қиймат бериш орқали.

Биринчи ва учинчи усуллардан фойдаланиш тавсия этилмайди. Сабаби, уларда HTML ва CSSда ёзилган кодлар аралашшиб кетади. Иккинчи усулда эса биз биргина файлни алмаштириш орқали сайтишимизнинг тўлалигича шамойилини ўзгартиришимиз мумкин. Шуни таъкидлаб ўтиш лозимки, айни бир объектнинг битта аломатига икки хил қиймат берилса, браузер охирги қиймат бўйича акслантиради. Шунинг учун учинчи усулда ёзилган буйруқ аввалгиларини эътиборсиз қолдиришни таъқозо этади ва қайсиидир ҳолларда бу имконият ҳам асқотиши мумкин.

Бир-бирига зиддиятли шартлар берилганда одатда «style» аломати ичидагиси инобатга олинса ҳам, алоҳида файл сифатида яратилган қонуниятларни қатъий тадбиқ этиш усули мавжуд. Бунинг учун берилган қийматдан сўнг «!important» жумласини ёзиш лозим. Масалан, қаерда кўрсатилган ранг қатъий тадбиқ этилиши лозим бўлса, у қуидагича ёзилади:

```
color: red !important;
```

Агар «!important» CSS файлда ҳам, тегнинг «style»ида ҳам ёзилган бўлса, албатта энг яқини, яъни «style» ичидагиси инобатга олинади.

Шу ўринда яна бир муҳим нарсани қайд этиб ўтиш лозим. Ҳар бир браузер учун фойдаланувчининг ўзи маҳсус қоидалар асосида яратиб олган CSS файлини юклаб олиб, уни барча веб-саҳифаларга бирдай тадбиқ

этиш мумкин. Бунда саҳифа орт рангини, хуснихатлар номи, ранги ва ўлчамини, баъзи тегларнинг акслантирилиш жилосини ва бошқа CSS стандартлари бўйича қонуниятларни белгилаш мумкин. Келгусида браузерга қўшимча юкланиб, барча саҳифалар учун тадбиқ этиладиган ташқи CSS файлни ва браузернинг ўзидағи созлашлар орқали акслантириш учун киритилган маҳсус ўзгартиришларни «фойдаланувчи жилоси» деб юритамиз. Веб-саҳифа яратувчи HTML ва CSS файллар ичida ёзиш орқали берилган қонуниятларни эса «муаллиф жилоси» деб атаемиз. Ҳар иккисида ҳам ўз қонуниятлари устивор ҳисобланиши учун «!important»ни қўллаш мумкин. Агар айни бир элемент учун берилган шартларнинг қайси бирида «!important» бўлмаса, ўз-ўзидан у эътиборсиз қолдирилади. Қуйида ҳар иккисида ҳам «!important» қўйилган ҳолларда айнан қайси бири инобатга олинишини жадвал тарзда кўриб чиқамиз, чунки фойдаланувчи ўзига хос шамойил яратаман деб, эътибор бермаган ҳолда дастурловчи томонидан ўрнатилган маҳсус шакларни бузиб юбориши мумкин:

Фойдаланувчи жилоси	Муаллиф жилоси
+	-
+	+
-	+
-	-

Бунда «+» ва «-» орқали ким томонидан «!important» қўлланилганлиги, катакнинг тўқлиги билан қайси бири инобатга олиниши қўрсатилган.

8.2. Умумий синтаксис

CSSда ҳам эълон қилиш бўлими мавжуд бўлиб, одатда уни ёзиш шарт эмас. Ушбу бўлимда қонуниятлар қандай ёзилганлиги ва қай тарзда тадбиқ этиш лозимлиги қўрсатилади.

Синтаксиси қуидаги:

@бүйрүк қонуниятлар

Мисол:

`@charset "utf-8";`

Изоҳ сифатида ёзилган ёки браузер эътибор қаратмаслик лозим бўлган ёзувлар маҳсус белгилар «/*» ва «*/» ларнинг орасига олиб қўйилади.

Умуман олганда CSS қуидаги синтаксис асосида баён этилади:

номланишлар { қоидалар }

Бунда, номланишлар:

номланиш [, номланиш]

қоидалар:

қоида [, қоида]

қоида:

хусусият: қийматлар

қийматлар:

қиймат [қиймат]

Эълон қилиш бўлимидаги «қонуниятлар» шаклан «қоидалар» кўринишида ҳам берилиши мумкин. Масалан, маҳсус хусниҳат қўшиш:

```
@font-face {
    font-family: Consolas; /* Хусниҳат номи */
    src: url(/font.ttf); /* Хусниҳат файлининг манзили */
}
```

Номланиш аслида келишилган идентификатор [2.6.] бўлиб, унда иккинчи белгисидан бошлаб чизикча (-) қатнашишига рухсат этилади. Бу номланишларни гурӯхлашда фойдаланилади. Номланиш қуидаги маънолардан бирини англашиб мумкин:

◆ **Универсал номланиш.** Саҳифадаги барча элементларга тааллуқли бўлиши учун номланиш сифатида юлдузчадан фойдаланилади, у барча номланиш маъносини беради, бальзи ҳолларда уни ишлатмаса ҳам бўлади. Мисол:

```
* { font-family: Times New Roman, sans-serif; }
```

◆ **Тегнинг номи.** Бунда саҳифадаги барча шу номдаги теглар кўрсатилган қоидалар бўйича жилоланади. Мисол:

```
div { color: grey; } <div>GREY</div>
```

◆ **Тегнинг `<id>`си.** Бунда фақат «`id`»си шу идентификаторга тенг бўлган обьект тусланади холос. CSSда бунақа номланиш идентификатор олдига «`#`» белгиси қўйиш билан бошланади. (Агар маълум тегнинг шу номдаги идентификатори учунгина ушбу қоидалар тадбиқ қилиниши лозим бўлса, тег номидан сўнг панжара ва идентификатор ҳеч қандай бўш жойларсиз номланиш сифатида ёзилиши керак. Лекин битта саҳифада идентификатор такрорланмас бўлиши талаб этилгани боис, бундай синтаксис деярли ишлатилмайди.) Мисол:

```
#mydiv { color: grey; } <div id="mydiv">GREY</div>
```

◆ **Синфлари.** Маълум ном билан HTML тегларини синфларга ажратиш мумкин. Бунинг учун керакли теглардаги универсал ҳисобланувчи class аломатига эълон қилинган номни берилади. Номланиш синфианглатувчи идентификатор бўлса, унинг олдига нуқта қўйилади. Мисол:

```
.myclass { color: grey; } <div class="myclass">GREY</div>
```

◆ Агар кўрсатилган қоидалар фақат маълум теглар учун тадбиқ этилиши талаб этилса, тегнинг номини синф идентификаторидан олдинги нуқтадан ҳам аввал бўш жойларсиз ёзиш керак: `p.myclass`. Агар обьектга бир пайтда бир неча синфларнинг қоидаларини умумлашмасини тадбиқ этиш лозим бўлса, «`class`» аломатида улар бўш жой билан ажратилиб кўштириноқ ичида ёзилади. Айнан

фақат бир неча синф күрсатылған тегларғагина тааллукли бўлиши учун номланиш кўйидагича күрсатилади:

```
.class1.class2 {color:grey;} <div class="class1 class2">GREY</div>
```

◆ Ичидағи теглар. Баъзан маълум тегларни бошқача қоидалар асосида туслашга тўғри келади, агар улар бошқа бир тегнинг ичидаги ётсагина. Яъни бир тегнинг ичидаги жойлашган бошқа тегларнинг туслаш учун уларнинг номлари кетма-кет бўш жой билан ажратиб ёзилади. Мисол:

```
div p { color: grey; } <div><p>GREY</p></div><p>TEXT</p>
```

◆ Қўшни тег. Айнан бир хил даражадаги «чуқурликда» жойлашган битта умумий ташқи тег ичидағилар қўшни теглар ҳисобланади. Масалан, «head» теги «body»га қўшни, лекин «body» теги «title»га қўшни эмас. Сабаби «html» асосий биринчи даражали тег ҳисобланса, унинг ичидағи иккинчи даражадагилар бир-бирига қўшни, учинчи даражали чуқурликдаги «title»га иккинчи даражадаги чуқурликдаги «body»га ҳам учинчи даражали «footer»га ҳам қўшни эмас. Қўшни теглар CSSда плюс (+) билан ифодаланади. Мисол:

```
div + p { color: grey; } <div>BLACK</div> <p>GREY</p>
```

Бунда қоида фақат плюсдан кейин күрсатылған номдаги тег(«p»)гагина тадбиқ этилади, агар «div» ва «p» қўшни теглар бўлиб, орасида бошқа элемент қатнашмаган бўлса.

◆ Қардош теглар. Улар айнан «қўшни тег» сингари аникланади, фақат ягона эмас, биринчи номланишдаги тегга нисбатан бир хил чуқурликда жойлашган ва ундан кейин келган иккинчи номланишга мос тегларнинг барчасига тадбиқ этилади. Қардош теглар «~» билан кўрсатилади:

```
div ~ b { color: grey; } <div>BLACK</div> <b>GREY</b>
<p><b>BLACK</b></p> <b>GREY</b>
```

◆ Ички теглар. «Ичидағи теглар»дан фарқли ўлароқ, «ички теглар» назарда тутилаётган тегнинг ичидаги жой-

лашади холос. Яъни таркибидаги бошқа тегларнинг ички қисмида бўлса, бу инобатга олинмайди. Кўзда тутилаётган тег n-дараражали чукурлиқда жойлашган бўлса, унга ички тег n+1 дараражада бўлиши шарт. Масалан, «meta»лар «head»га ички теглар бўлиши шарт, «div» эса «body»га нисбатан бундай маъсулиятни олмаган. Ички теглар CSSда »» белгиси билан ифодаланади. Мисол:

<code>div > p { color: grey; }</code>	<pre><div> <p>GREY</p> <p>BLACK</p> </div></pre>
--	---

◆ **Аломатли.** Тегларнинг аломати ва у қабул қилган қийматига қараб жилолаш учун тўғри қавсдан фойдаланилади. Бундай усулда жило беришнинг бир қанча кўринишлари мавжуд:

<u>номланиш[аломат]</u> {...}	«аломат»и (қиймати аҳамиятсиз) кўрсатилганлар учун
<u>номланиш[аломат=қиймат]</u> {...}	«аломат»и «қиймат»га тенг бўлганлар учун
<u>номланиш[аломат^=қиймат]</u> {...}	«аломат»и «қиймат» жумласидан бошланадиганлар учун
<u>номланиш[аломат\$=қиймат]</u> {...}	«аломат»и «қиймат» жумласи билан якунланадиганларга
<u>номланиш[аломат*=қиймат]</u> {...}	«аломат»и «қиймат» жумласига эга матндан иборатларга
<u>номланиш[аломат~=қиймат]</u> {...}	«аломат» таркибида (бўш жой билан) «қиймат» мавжудларга
<u>номланиш[аломат =қиймат]</u> {...}	«аломат»нинг «қиймат»ида чизикча (-) бўлганлар учун

Мисоллар:

<code>p[align] { font-weight: bold; }</code>	<code><p align="right">BOLD</p></code> <code><p>REGULAR</p></code>
<code>p[align="center"] { font-weight: bold; }</code>	<code><p align="center">BOLD</p></code> <code><p align="left">REGULAR</p></code>
<code>p[title^="c"] { font-weight: bold; }</code>	<code><p title="center">BOLD</p></code> <code><p title="left">REGULAR</p></code>

p[align\$="t"] {font-weight: bold; }	<p align="right"> BOLD </p>
p[align*="cent"] { font-weight: bold; }	<p align="center"> BOLD </p>
p[class="bir"] { font-weight: bold; }	<p class="bir nom"> BOLD </p>
p[align!="bir"] { font-weight: bold; }	<p class="nom"> REGULAR </p>
	<p class="bir-nom"> BOLD </p>
	<p align="bir"> REGULAR </p>

Бир нечта тенг кучли номларни ёзиш лозим бўлса, уларни юқоридаги синтаксисда келтирилгани каби вергул билан ажратилади. HTMLда эса шунчаки бўш жой ташлаб ёзилар эди. Масалан:

```
.my-p, .my-div { color: grey; } <div class="my-div my-p">GREY</div>
```

Шу ўринда нозик бир нуқтага эътибор қаратиб кетиш лозим. Агар бир неча синф кўрсатилган бўлса, обьект улардаги қоидаларнинг умумлашмаси сифатида аксланади. Айтайлик, айни бир аломатга келтирилган икки синфда турлича қиймат берилган бўлса, охирги кўрсатилган номдаги синфдаги қоидалари эмас, балки CSSда охирги эълон қилинган синф қоидалари тадбиқ этилади. Юқоридаги мисол бўйича, CSSда аввал «my-p» эълон қилинган бўлса, «class»да қандай кетма-кетликда берилишидан қатъий назар аввал «my-p» асосида акслантирилиб, сўнг «my-div» қоидалари тадбиқ этилади.

8.3. Қийматларнинг турлари

CSSда ҳам қийматлар HTMLда қўлланиладиган аксарият турларни қабул қиласди. [2.6.] Фақат айрим ўзига хос қўшимчалар киритилган.

- **inherit** — қийматни ташки тегнинг айни хусусиятидан олади. Агар ташки тегга мос қиймат ўрнатилмаган бўлса, одатий қийматни қабул қиласди. Таъкидлаб ўтиш лозимки, тег учун кўрсатилган хусусиятларнинг

қийматлари одатда унинг ичидаги тегларга ҳам таъсир қиласи, шундай экан, ушбу қийматни бериш кўпинча самарасиз. Аммо бაъзан қийматлар қўшини теглар хусусиятларига қараб ўзгарамади, шундай ҳолларда айрим хусусиятлар ташки тегники сингари бўлишини аниқ кўрсатиб ўтишга тўғри келиши мумкин.

- **initial** — асл қийматини бериш. Юқорида таъкидланганнидек, ичкаридаги теглар одатда ташқисидагидан хусусиятларни ўзлаштириб олади. Лекин баъзи тег учун унинг кўрсатилган хусусиятига браузерда белгиланган асл қиймат бериш лозим бўлса, ушбу сўз ишлатилади.
- **unset** — агар ташки тегнинг мос хусусиятига қиймат берилган бўлса, «*inherit*» сингари уни қабул қиласи. Акс ҳолда «*initial*» сифатида ишлайди.

Ушбу учала қийматлардан бирини қабул қиласиган **all** хусусияти аввалдан мавжуд бўлган катта CSS файлларни ўзгартирмасдан ундаги баъзи қоидаларни бекор қилиш учун кенг қўлланилади. Айрим ҳолларда мавжуд йирик лойиҳаларга кичик ўзгартирishiшлар қўшиш ҳам талаб этилади. Шундай вазиятларда олдиндан киритилган жилоларни ўрганиб чиқиб ўзгартирishiшга ваqt кўп сарфланиши мумкин, «*all*» хусусити эса буни осонгина ҳал эта олади:

```
h3 { all: unset; color: blue; }
```

Рангнинг қабул қилувчи қийматлари HTMLницидан анча мукаммалроқ. Умуман олганда рангни фақат CSS орқали бериш керак, бу объектга жило беришга киради ва фақат алоҳида файлда кўрсатилиши мақсадга мувофиқ. Аслида шакл-шамойилни аниқловчи катталикларнинг ҳаммаси CSS ёрдамида аниқланиши тавсия этилади, лекин бу ҳар доим ҳамма дастурловчиларга маъқул келавермайди. Шундай бўлса-да, тусловчи омиллар, айниқса — ранг, фақат CSSда кўрсатилиши қатъий талаб этилади.

Ранг номи ва RGB [2.6.] кодланишдан ташқари, қийматни **rgb** ва **rgba** функциялари [8.4.] билан ҳам бериш мумкин. Бунда рангларнинг ўнлик саноқ тизимидағи

мос қийматлари ҳамда қай даражада шаффоғлигини күрсатилади. Синтаксиси:

<code>rgb(x,x,x)</code>	x — 0..255 оралиқдаги бутун сон ёки унинг фоиздаги ифодаси
<code>rgba(x,x,x,a)</code>	a — 0..1 оралиқдаги шаффоғликни билдирувчи ҳақиқий сон

HSL (Hue – соя, Saturate – түклик, Lightness – ёрқинлик) функцияси ёрдамида рангларнинг кодларини айланып сифатида жойлаштириб «рангли доира» ҳосил қилинади ва мос бурчакдаги қийматни RGB тартибида күрсатиш билан таниланади. Бунда ҳар бир сон 0..359 орасыда ёки шунга нисбатан фоиз ҳисобида берилади. Синтаксиси:

`hsl(x,x,x)` x — 0..255 орасидаги бутун сон ёки 0%дан 100%гача

Бунга шаффоғлик омили құшилиб `hsla` функцияси ҳосил қилинган. Синтаксиси:

<code>hsla(x,x,x,a)</code>	x — 0..255 оралиқдаги бутун сон ёки фоиз ҳисобида (0%дан 100%гача); a — 0..1 орасида
----------------------------	--

Булардан ташқари ранг фойдаланувчи созлашларидан олинниши мумкин. Агар келтирилган номга мос ранг топтимаса, вазифасига күра энг яқын элемент ранги олинади. Бундай усул яратилаёттан обьектлар рангини операцион тизимде мавжудларига мос ҳолда олиш билан саҳифа тусини тез ва мутаносиб равища үзгартыришта катта ёрдам беради. Ушбу усул айни веб-саҳиғаны түрли браузер ва компьютерларда турлича хилда намоён бўлишини таъминлайди. Ҳар кимга ҳар хил ранг ёқади, фойдаланувчининг қурилмасида кўзи ўргангандан рангларда сайтнинг намоён бўлиши, унга нисбатан эътиборни жалб қилишнинг бир ижобий омилидир. Шу ўринда мавжуд калит сўзлар рўйхатини келтириб ўтамиз:

- `activeborder` — фаол ойна чегарасининг ранги.
- `activecaption` — фаол ойна сарлавҳасининг ранги.
- `appworkspace` — кўп хужжатли ойна орқа тасвири.
- `background` — ишчи стол орқа тасвири.
- `buttonface` — ҳажмли обьектнинг олд томони ранги.

- **buttonhighlight** — ҳажмли объектнинг тўйинган сояси ранги.
- **buttonshadow** — ҳажмли объектниг сояси ранги.
- **buttontext** — тутмадаги матн ранги.
- **captiontext** — сарлавҳадаги матн ранги.
- **graytext** — ўчирилган (нофаол) матн ранги.
- **highlight** — белгиланган объект ранги.
- **highlighttext** — белгиланган объект матнининг ранги.
- **inactiveborder** — нофаол ойна чегарасининг ранги.
- **inactivecaption** — нофаол ойна сарлавҳасининг ранги.
- **inactivecaptiontext** — нофаол сарлавҳа ранги.
- **infobackground** — сузувчи изоҳ орқа тасвири ранги.
- **infotext** — сузувчи изоҳ матнининг ранги.
- **menu** — меню орқа тасвири ранги.
- **menutext** — меню матнининг ранги.
- **scrollbar** — айлантиргич ранги.
- **threaddarkshadow** — ҳажмли элемент тўқ сояси ранги.
- **threedface** — ҳажмли элементнинг олд томони ранги.
- **threedhighlight** — белгиланган ҳажмли элемент ранги.
- **threelightshadow** — ҳажмли элемент оч сояси ранги.
- **threedshadow** — ҳажмли элементнинг тўйинган сояси ранги.
- **window** — ойна (орқа тасвири) ранги.
- **windowframe** — ойна ҳошиясининг ранги.
- **windowtext** — ойнадаги матн ранги.

8.4. Функциялар

CSSда қийматларнинг нафақат келишилган ўлчов бирликлари ёрдамида, балки маҳсус функциялар орқали ҳам бериш мумкин.

Шу ўринда функция нималиги ҳақида тушунчаси бўлмаган муштариyllар учун эслатиб ўтишни жоиз топдик.

Маълум амаллар, ҳисоб-китоблар бажариб, қайта ишлаш жараёнидан сўнг маҳсус натижа қайтарадиган дастурнинг кичик бўлагига функция дейилади. Унга қўшимча равишда кириш қийматлари ҳам берилиши мумкин.

Юқорида биз берилган сонларга мос рангнинг қийматини қайтарадиган «`rgb`», «`rbga`», «`hsl`» ва «`hsla`» каби функциялар билан танишдик. Энди тасвириңнинг күринишини ўзгартирадиган маҳсус функцияларни кўриб ўтамиз.

8.4.1. Ҳиралаштириш

Объектнинг акслантиришда маълум функциялар ёрдамида күринишини ўзгартириш мумкин. Улар `filter` хусусиятига маҳсус ном ва қийматлар билан кўрсатилади. Объектнинг барча элементларини (Гаусс усулида) ҳиралаштириб кўрсатиш учун `blur` функцияси қўлланилади. Синтаксис:

```
filter: blur(уличам);
```

Ушбу функцияни «`body`»дан ташқари ихтиёрий кўринадиган тег учун қўллаш мумкин. Мисол:

```
<div class="xira">
    
    <span>Рақами: xxxxxxx</span>
</div>
<p>Номаълум шахс сурати</p>
```

```
.xira {
    filter: blur(8px);
}
```

8.4.2. Ёрқинлаштириш

Тасвиrlарни ёрқинлигини ошириш ёки камайтириш учун `brightness` функциясидан фойдаланилади. Оддий ёзувларга бу усулни қўллаш кутилган самара бермайди. Расмнинг оддий ҳолатида ёрқинлик 1 ёки 100% деб хисобланади ва шунга нисбатан фоизда ёки ҳақиқий сон билан қиймат берилади. Қиймат манфий бўлиши мумкин эмас, агар кўрсатилмаса 1 деб хисобланади ва ҳеч нарса ўзгармайди. Синтаксис:

```
filter: brightness(фоиз);
```

8.4.3. Хижиллаштириш

Тасвири ёрқинлаштиришга қарама-қарши амал — хижиллаштиришдир, шунинг учун ҳам уни зиддиятлаш маъносидаги «*contrast*» сўзи билан номланган. Синтаксиси ва унга бериладиган қиймат айнан ёрқинлаштиришники сингари.

8.4.4. Тўйинтириш

Расм рангларнинг бир-бирига яқинини бирлаштириш учун *saturate* функцияси ишлатилади. Унга 0 қиймат берилса, сурат оқ-қора кўринишга келиб қолади. 1 қиймат берилганда эса, ўзгаришсиз қолади. Унга қанча катта қиймат берилса, шунча расмдаги ранглар сони камайиб боради. Синтаксис:

```
filter: saturate(ҳақиқий сон);
```

8.4.5. Шаффофлик

Тасвирларни шаффофлигини таъминлаш учун *opacity* функцияси ишлатилади. Шаффоф обьект ўз рангини йўқотиши билан бирга орқасидаги тасвирларни кўришга имкон яратади, яъни тиник ойна (кўзгу эмас) тасаввурини беради. Синтаксис:

```
filter: opacity(фоиз);
```

8.4.6. Соя қўшиш

Тасвирларни жилолашда уларнинг кўринишидаги таъсирини ошириш учун соя қўшиш ҳам муҳим аҳамиятга эга бўлади. Бунда обьект — хусусан расм, тўғри тўртбурчак шаклида қиёфага эга бўлиши шарт эмас, соя унинг чегараси бўйлаб жойлашаверади. Синтаксис:

```
filter: drop-shadow(x, y, r, c);
```

Бунда, *x* – соянинг уфқий сурилиши, киритилиши мажбурий қиймат. Мусбат сон бўлса ўнгта, манфий бўлса чапга кўчади; *y* – соянинг тиккасига сурилиши, маж-

бурий қиймат. Мусбат бўлса пастга, манфий бўлганда тепага қўчади; **г** – соянинг суваштириш радиуси, қиймат берилмаса 0 қабул қилинади ва бунда суваштирилмаган тиник ҳолда кўринади; **с** – соя ранги. Мисол:

```
<div class="soya">
  
  <span>Олма</span>
</div>
```

8.4.7. Оқ-қора тасвир

Тасвирни оқ-қора тусга киритиш (монохромлаш) учун **grayscale** функцияси қўлланилади. Унга фоиз ҳисобида қиймат берилади. Агар 1 ёки 100% қиймат берилса, расм тўлалигича оқ-қора кўринишга ўтади, қиймат берилмаса, 0 деб қабул қилинади ва тасвир ранги ўзгармайди. Синтаксис:

```
filter: grayscale(фоиз);
```

8.4.8. Расм рангини ўзгартириш

Ёзувларнинг рангини ўзгартириш жуда содда, лекин баъзан расмнинг рангини ҳам бошқача тусда чиқаришга тўғри келади. Бу билан айни бир суратни қайта-қайта сақлаб хотирани беҳуда сарфлаш олди олинади. Расмнинг рангини бутунлай ўзгартириш учун «рангли доира»га [8.3.] асосланган **hue-rotate** функцияси қўлланилади. Синтаксис:

```
filter: hue-rotate(бурчак);
```

Объектдаги аксланаётган ҳар бир нуқта ўз рангига эга. Ўша рангнинг кодининг ўрнига максимал қийматта нисбатан тескарисини берилса, тасвир фотокамера пленкасида кўринадигандек тус олади. Шундай усуслни рангни ағдариш деб ҳам юритилади, бундай имкониятни **invert** функцияси тақдим этади. Синтаксис:

```
filter: invert(фоиз);
```

Қадимий суратлар нафақат оқ-қора рангда, балки жи-гарранг тусга ҳам эга күринишида бўлгани учун, айнан шундай эски тасвир таасуротини беришида **sepia** функциясидан фойдаланилади. Синтаксис:

```
filter: sepia(фоиз);
```

8.4.9. Орқатасвир ранглари

CSS ёрдамида объект орқасига расм қўйиш учун **url** функциясидан фойдаланган ҳолда **background-image** ху-сусияти ишлатилади. Синтаксис:

```
background-image: url(ишорат);
```

Ишоратда расмнинг манзили кўрсатилади. Орқа тас-вирнинг қай тарзда жойлаштириш ҳақида қўйида ба-тағсил тўхтalamиз, ҳозир ушбу хусусиятга яна қандай функциялар ёрдамида қиймат бериш мумкинлиги билан танишиб ўтамиз.

▪ Чизиқли тусланиш.

```
background-image: linear-gradient(йўналиш, ранг1, ранг2);
```

Бунда объектнинг орқа қисми «ранг1»дан «ранг2»га қараб ўзгариб боради, агар «йўналиш» кўрсатилган бўлса, айнан у бўйича, акс ҳолда тепадан пастга тик равишида ранглар алмашинилади. Йўналиш «бурчак» ҳисобида [2.6.] ёки қўйидаги инглизча номланишлар билан берилиши мумкин:

- **to top** – пастдан тепага, тик (0deg)
- **to left** – ўнгдан чапга, ётиқ (270deg ёки -90deg)
- **to bottom** – тепадан пастга, тик (180deg)
- **to right** – чапдан ўнгта, ётиқ (-270deg ёки 90deg)
- **to top left** – пастки ўнг бурчакдан тепа чапга, ёнбош
- **to top right** – пастки чап бурчакдан тепа ўнгта, ёнбош
- **to bottom left** – тепа ўнг бурчакдан пастки чапга, ёнбош

- to bottom right – тепа чап бурчакдан пастки ўнгга, ёнбош

- Ёйсимон тусланиш.

background-image: radial-gradient(*шакл* *услуб* at *жойи* *ранглар*);

Бунда «шакл»нинг одатий қиймати эллипс бўлиб, унга аниқ «circle» ёки «ellipse» қиймати берилиши мумкин, «услуб»да шаклнинг радиуси берилади ёки қандай ўлчамни эгаллаши кўрсатадиган қуйидаги сўзлардан фойдаланилади:

- closest-side – «шакл»нинг чегараси ўзи жойлашган майдоннинг энг яқин томонига тегадиган ҳолатда чизилади. Эллипс бўлса, иккита яқин томонлар инобатга олинади. Бунда шакл тўлалигича аксланиб, ташқи обьект ичига жойлашади.
- closest-corner – «шакл» чегараси майдоннинг энг яқин бурчагига тегадиган бўлади.
- farthest-side – чегара майдоннинг узоқ томонлариға нисбатан олинади.
- farthest-corner – чегара майдоннинг узоқ бурчагига тегадиган тарзда чизилади.

Шаклнинг маркази қаерда бўлишини «жойи»га уфқий ва тик йўналишлар бўйича пиксел ёки фоиз ҳисобида қиймат бериш билан кўрсатилади. Албатта, «жойи» стандарт ўринлардан бирини эгаллаши лозим бўлса, қийматни «top», «bottom», «right», «left», «center» каби инглизча сўзлар ёрдамида берса ҳам бўлади.

Нечта ранг ўзгариб бориши орқали акслантириш лозим бўлса, вергул билан ажратилган ҳолда шунча «ранглар» бериш мумкин. Мисол:

background-image: radial-gradient(circle farthest-corner at top left, cyan, aqua);

- Такрорий тусланиш.

Майдон юзаси катта бўлганда тусланишда унинг кўп қисми айни биттга ранг билан тасвиirlаниб қолади. Бундай пайтларда қонуниятни такрорлаш яхши самара беради. Бўнинг учун чизиқли ва ёйсимон тусланиш-

ларни тақрорлаб акслантирадиган махсус `repeating-linear-gradient` ва `repeating-radial-gradient` функциялар мавжуд. Уларга бериладиган қийматлар мос функцияси-ники билан бир хил. Фақат тақрорланишни таъминлаш учун ўзгариб борадиган, яъни иккинчи рангдан бошлиб кейингиларига қанча қалинликни эгаллаши кўрсатилиб борилиши лозим. Мисол:

```
background-image: repeating-radial-gradient(  
    circle at top left, cyan, aqua 10px, blue 80px  
) ;
```

8.4.10. Ҳисобланган қиймат

Жуда кўп ҳолларда объектларнинг ўлчамини ва жойлашиш ўрнини кўрсатиш учун ташқи тегнинг акслangan майдони ҳажмини инобатта олишга тўғри келади. Фоизларда қиймат бериш доим ҳам кутилган самара бермайди. Бундай вазиятларда махсус calc функцияси ёрдамида ҳисобланадиган қийматни хусусиятга беришга тўғри келади. Унда тўрт арифметик амал: қўшиш, айриш, кўпайтириш ва бўлишларни амалга ошириш мумкин. Унутмаслик лозимки, «+» ва «-» ишоралари ҳар икки томондан бўш жой билан ажратилиши шарт! Акс ҳолда браузер уни тушунмайди. Умуман олганда барча амалларни бўш жой билан ажратиб ёзиш тавсия этилади. Мисол:

```
width: calc(100% - 20px);  
height: calc(100% / 2);
```

8.5. Танлов асосида бириткириш

Аввал, веб-саҳифа намоён бўладиган қурилма (media) турларини эслаб ўтсак [2.4.]:

- **all** – барча қурилмалар
- **aural** – товушли браузерлар
- **braille** – Брайл тизимли қурилмалар (кўрлар учун)
- **handheld** – уяли (кичик экранли) қурилмалар
- **print** – Чоп этиш қурилмаси

- **projection** – Проектор
- **screen** – Монитор
- **tv** – Телевизор

Юқорида келишиб олганимиздек, CSS қонуниятларини алоҳида файлга ёзамиз. Зарур бўлганда керакли саҳифага айнан керакли «css»ни юклаб фойдаланаверилади. Бундан ташқари битта «css» файл ичидаги бошқасини чақириб бирлаштириб олиш имконияти ҳам мавжуд. Бунинг учун қуийдагича синтаксис қўлланилади:

```
@import "«css» файл манзили" медиа тури;
```

Мисол:

```
@import "/style/main.css" screen;
/* компьютерлар учун */
@import "/style/smart.css" print, handheld;
/* чоп этиши ёки смартфонлар учун */
```

Нафакат файлларни саралаш асосида бириткириш, балки қонуниятларнинг ўзини ҳам танлов бўйича ёзиш мумкин. Мисол:

```
@media screen { /* компьютерлар учун */
    BODY {
        color: darkblue; /* матн ранги тўқ кўк */
    }
}
@media print { /* чоп этиши учун */
    BODY {
        color: black; /* матн ранги қора */
    }
}
```

8.6. Табақалаш

Одатда CSS қоидалари HTML тегларини мъълум синтаксисга ажратиб, шулар бўйича тадбиқ этиши асоси-

да ишлайди. [8.2.] Лекин бундай тақсимлаш доим ҳам етарли эмас. Масалан, ўнлаб ишоратлар рўйхатига эга веб-саҳифани қарайдиган бўлсак, яна қайта босиб вақт сарфламаслик учун, кўп ҳолларда, қайсилариға ташриф буюрилганлигини ажратиб кўрсатиш талаб этилади. Бунга ёрқин мисолни глобал излаш хизматларини тақдим этувчи сайtlарда кўрганмиз (google.com, yandex.ru ва бошқалар).

Айни бир хил синфга оид объектларни жорий ҳолатига қараб бир неча табақага ажратиб, ҳар бирига ўзига мос CSS қоидалар бериш мумкин. Бунинг учун «номланиш»дан [8.2.] сўнг икки нуқта (:) ва ҳолат номи бўш жойларсиз ёзилади.

◆ **:active** — объектнинг фаоллашган ҳолати. Курсор билан танланиб, сичқончани босиб турилгандаги (ҳали қўйиб юбормасдан аввалги) ҳолат. Мисол:

```
div:active { color: maroon };
```

◆ **:checked** — «radio» ва «checkbox» дарчаларининг белгиланган ҳолати.

◆ **:default** — айнан бир хил омиллар билан берилган теглар ичидан одатий қиймат сифатида қабул қилинган элементнинг ҳолати. Масалан, битта форма ичида бир неча юборувчи «submit» тугмалари қўйилган бўлса, киритиш («Enter») босилганда одатда уларнинг биринчиси ишга тушади, яъни у «:default» ҳолатида бўлади.

◆ **:disabled** — нофаол ҳолат. Фойдаланувчи бундай элементлар қийматини ўзгартира олмайди, тугма бўлса, босиш имконсиз, яъни ўз фаолиятини бажармайди. Бундай объектлар одатда хиралаштириб кўрсатилади, яна қандайдир қўшимча жило бериш лозим бўлса, ушбу табақадан фойдаланилади.

◆ **:empty** — тегларнинг ичида ҳеч қандай элемент ва ҳаттоқи бўш жой ёки кўринмайдиган белгилари ҳам бўймаган ҳолати.

◆ **:enabled** — нофаоллаштирилмаган элементлар учун.

◆ **:first-child** — ташқи тегнинг ичидаги биринчи жойлаш-

ган элемент учун. Бундай ҳолат бир нечта қўшни объектларнинг биринчиси учун бошқачароқ жило бериш лозим бўлганда ас қотади.

◆ **:last-child** — ташқи тегдаги энг охирги элементтаги на тадбиқ этилади.

◆ **:first-of-type** — ташқи тег синфига оид барча тегларнинг биринчи элементи учун тадбиқ этилади. Масалан, жадвал яратилаётганда ҳар бир катақни англатувчи «td» теги «tr»нинг ичидаги ётади. Номланиш «td:first-of-type» кўринишида берилса, ҳар бир қатордаги биринчи катак учун қўлланилгани боис, жадвалнинг биринчи устуни кўрсатилган ўзгача жило бўйича аксланади.

◆ **:last-of-type** — ташқи тег синфидағи барча теглар ичидаги охирги элементта тадбиқ этилади.

◆ **:focus** — фокус қабул қилган элемент ҳолати, яъни «tab» тугмасини босиш орқали танланган элемент учун тадбиқ этилади.

◆ **:fullscreen** — ушбу ҳолат браузер тўлиқ экранга ёйилган ҳолда акслантирилади.

◆ **:hover** — «сичқонча» кўрсаттичи обьект устига борган лекин ҳали босилмаган ҳолат. Шуни унутмаслик керакки, ушбу ҳолатнинг таъсири «:active» бўлганда ҳам йўқолмайди. Яъни обьектни сичқонча билан босилар экан, албатта кўрсаттич (курсор) унинг устида бўлади.

Шундай хусусият мавжудки, у фақат кўрсаттич обьектнинг устига келганда акслантирилади. Уни «:hover»да бериш шарт эмас, масалан: **cursor**, қиймати аниқланган тур:

- url — кўрсаттич расмининг манзили берилади;
- auto — кўринишини обьект турига қараб ўзи аниқлайди;
- none — кўрсаттични яширади;
- default, context-menu, help, pointer, progress, wait, cell, all-scroll, crosshair, text, vertical-text, alias, not-allowed, no-drop, copy, col-resize, row-resize, n-resize, ne-resize, e-resize, se-resize, s-resize, sw-resize, w-resize, nw-resize, nesw-resize, nwse-resize, zoom-in, move, zoom-out, grabbing, grab — оғерацион тизимда аниқланган мот расмларни акслантиради;

◆ **:in-range** — объектга берилган қиймат унинг «min» ва «max» аломатларида кўрсатилган оралиқда бўлган ҳолат.

◆ **:out-of-range** — объектга фойдаланувчи берган қиймат унинг «min» ва «max» аломатларида кўрсатилган оралиқдан ташқарида бўлган ҳолат. Мисол:

```
<input type="number" value="30" max="359" min="0"  
placeholder="Бурчакни киритинг">
```

```
input:in-range { background: #e2edc1; }  
input:out-of-range { background: #f9c3d2; }
```

◆ **:indeterminate** — белгиланувчи дарчанинг ноаниқ ҳолати, яъни «checkbox» қиймати рост ҳам ёлғон ҳам эмас.

◆ **:invalid** — киритилган қиймати дарчанинг турига мос бўлмаган ҳолат. Масалан, почта манзили ўрнига оддий ишорат киритилса.

◆ **:lang** — тегнинг «lang» аломатида кўрсатилган тил мос келган ҳолат. Мисол:

```
<q lang="ru">Надежный источник</q>  
<q lang="uz">Муқаммал қўлланма</q>
```

```
q:lang(ru) { color: darkred; }  
q:lang(uz) { color: maroon; }
```

◆ **:link** — ҳали ташриф буюрilmаган ишорат ҳолати.

◆ **:not** — аломатининг қиймати кўрсатилгандаидек бўлмаган элементлар учун тадбиқ этилади. Мисол:

```
<input type="button" value="Босиш">  
input:not([type="submit"]) { color: maroon; }
```

◆ **:nth-child(қиймат)** — Кўрсатилган тегнинг ичидаги элементларни тартиб бўйича саналганда, n-ларига тадбиқ этилади. Бунда «қиймат» қуидагилардан бирини қабул қиласди:

○ **odd** — барча тоқ ўриндаги элементлари учун;

○ **even** — барча жуфт ўриндаги элементларга;

○ бутун сон берилганда, ўша ўринда жойлашган элементга;

○ «n» қатнашган арифметик ифода берилганда, унда

н натурал сон сифатида саналиб, ўша ифода қиймати бўйича ўринни эгаллаган элементга; Мисол:

```
tr:nth-child(2n) { background-color: #f0f0f0; /* ортранги */ }
```

◆ **:nth-last-child** — ушбу ҳолат «:nth-child» сингари бўлиб, фақат саноқ охирги элементдан тескари тартибда амалга оширилади. Жадвалдаги қаторларни бир-биридан фарқлаб туриш учун бу усулни қўллаш яхши натижা беради.

◆ **:nth-of-type** — ташқи тег синфи дагиларнинг n-элементлари учун қўлланади. Бунда саноқ тартибини «order» хусусияти билан ўзгартириш мумкин.

◆ **:nth-last-of-type** — тескари саноқ бўйича «:nth-of-type»ни тадбиқ этилади.

◆ **:only-child** — кўрсатилган синфда ташқисининг ичидага ягона бўлган тег ҳолати.

◆ **:only-of-type** — ташқи синфга оид тегларнинг ичидага кўрсатилгандек ягона бўлган ҳолатларда тадбиқ этилади.

◆ **:optional** — «required» аломати кўрсатилмаган дарчалар учун қўлланилади.

◆ **:read-only** — «readonly» аломати берилган, ўзгартириш мумкин бўлмаган ҳолат.

◆ **:read-write** — қийматини ўзгартириш мумкин бўлган дарчалар учун.

◆ **:required** — «required» аломати кўрсатилган дарчаларга тадбиқ этилади.

◆ **:target** — «id»си веб-саҳифанинг манзилини кўрсатувчи ишпоратда «#»дан сўнг берилган идентификаторга тенг бўлган теглар учун қўлланилади. Мисол:

URL	http://yurt.uz/?a=4#hikoyalar
CSS	<code>h2:target { font-weight: bold; }</code>
HTML	<code><h2 id="sherlar"> Шеърлар </h2></code> <code><h2 id="hikoyalar"> Ҳикоялар </h2></code> <code><h2 id="hikmatlar"> Ҳикматлар </h2></code>
Натижа	Шеърлар Ҳикоялар Ҳикматлар

◆ **:valid** — фойдаланувчи томонидан киритилган қиймати аниқланган турга ва чекловга мос тушган дарчанинг ҳолати.

◆ **:visited** — фойдаланувчи томонидан ташриф буюрилган ишоратлар ҳолати.

8.7. Ҳаракатлантириш

Бир пайтлар «Internet Explorer» браузери югирувчи қаторни акслантириш учун **marquee** тегини ўзига хос аломатлар билан киритган эди. Кейинчалик бошқа браузерлар ҳам уни ўзларига тадбиқ эта бошлиди. Эътибор қаратиш керакки, ушбу тег HTMLда мавжуд бўлмаган ва HTML5 да эса уни қўллаш мутлақо тавсия этилмайди. Аммо бундан ҳам анча мукаммал тарздаги ҳаракатларни ифодалаш учун CSSнинг ўз имкониятлари етарли.

8.7.1. Ҳаракатлантирувчи хусусиятлар

Ҳаракатни бошқариш учун **«animation»** хусусиятидан фойдаланилади. Унинг кенгайтирилган имкониятлари билан батафсил танишамиз:

- **Ном бериш – *animation-name: ном***

Бунда «ном» идентификатор бўлиб, унга мурожаат қилишни таъминлайди. Агар ном ўрнида «none» қиймат берилган бўлса, ҳаракатланишни ўчиради. Айни бир қоидалар учун турли хилда ҳаракатланиш пайдо қилиш керак бўлса, «ном»га бир неча идентификатор вергул билан ажратилган ҳолда берилади ва уни ҳаракатланишга боғланганда мос номга мурожаат қилинади.

- **Бажарилиш вақти – *animation-duration: вақт***

Бошланғич ҳолатдан охиргисига қанча вақтда ўтиш лозимлигини кўрсатади, «ном» бир нечта бўлганда ҳар бири учун алоҳида вергул билан ажратиб вақт бериш мумкин. одатий қиймати: «0s», нол сония, ҳаракат сезилмайди.

- **Ҳаракат тезлиги – *animation-timing-function: қиймат***

Ҳаракатланишнинг бир маромдаги тезликда бажарилади.

лиши доим ҳам яхши самара бермайди. Эътибор жалб этиладиган вазиятда секинлашиб, кейин объектнинг тезроқ кўриниш соҳасидан четлатилгани маъкул. Бунинг учун физика фанидан бизга таниш бўлган тезланиш тушунчаси ишлатилади, аммо унинг ҳам ўзгармас бўлиши кутилган натижани бермайди, шунинг учун Безиер алгоритми бўйича уни ўзгартириб борилади. [6.4.1.] **cubic-bezier(x_1, y_1, x_2, y_2)** функцияси кўрсатилган нуқта ва координаталар боши бўйича ҳосил қилинган тўғри тўртбурчак ичига чизиладиган ёй ўзгариши бўйича қиймат олади. Баъзи стандарт тезлик ўзгаришларини маҳсус номлар билан ҳам бериш мумкин:

- **ease** – одатий қиймат, ҳаракат секин бошланиб, кейин тезлашади ва охирида яна секинлашади. Бу **cubic-bezier(0.25, 0.1, 0.25, 1)** функцияning қийматига тенг.
- **ease-in** – секин бошланиб, охирида тезлашади, **cubic-bezier(0.42, 0, 1, 1)** сингари.
- **ease-out** – тез бошланиб, охирида секиланапи, **cubic-bezier(0, 0, 0.58, 1)** сингари.
- **ease-in-out** – бошланиш ва якунида секинлашади, **cubic-bezier(0.42, 0, 0.58, 1)**дек.
- **linear** – ҳаракат бошидан охиригача бир хил тезликда бўлади.
- **step-start** – ҳаракатланиш бўлмайди, дарҳол якуний жойлашувга ўтади.
- **step-end** – ҳаракат бўмайди, кўрсатилган вақт турагач якуний нуқтага кўчади.
- **steps (сон, start | end)** – узлуксиз ҳаракат бўлмайди, кўрсатилган «сон» миқдорида сакрапшлар амалга оширилади. Бунда бошлангич ва охирги жойлашув нуқталари орасидаги кесмани тенг миқдорда «сон»та қисмга ажратилади ва ҳосил бўлган нуқталарда объектни кетма-кет кўчириб ўтилади. Агар бўлингандан бутун сон чиқмаса, «start» сўзи ёзилган ҳолда ҳосил бўлган ҳақиқий соннинг бутун қисми олинади, «end» деб кўрсатилганда эса ундан бир бирлиқ қаттаси олинади.

- Кутиш вақти – **animation-delay: вақт**

Кўрсатилган вақтдан кейин ҳаракат бошланади. Агар «ном» бир нечта бўлса, уларнинг ҳар бирига мос бошланғич кутиш вақтини вергул билан ажратилган ҳолда бериш мумкин. Вақт сония ёки миллисония ҳисобида берилади. [2.6.]

▪ **Такрорланиш сони –**

animation-iteration-count: *infinite* | *son*

- *infinite* – ҳаракатланиш тўхтовсиз давом этади.

○ «сон» неча марта ҳаракатланиш бажарилишини кўрсатади. Манфий қиймат бериш мумкин эмас. Агар 0.5 кўрсатилса, ҳаракатланишнинг ярми бажарилади.

▪ **Йўналиши –**

animation-direction: *normal* | *alternate* | *reverse* | *alternate-reverse*

- *normal* – ҳаракатланиш якунлангач, дарҳол бошланғич ҳолатга ўтади.

- *alternate* – ҳаракат якунлангач, тескари йўналишда ортга қайтади.

- *reverse* – ҳаракатланиш тескари йўналишда амалга оширилади.

- *alternate-reverse* – ҳаракат тескари йўналишда бошланади ва якунлангач ортга қайтади, яъни кўрсатилган йўналишда бажарилади.

▪ **Ҳаракат якунланиши –**

animation-fill-mode: *none* | *forwards* | *backwards* | *both*

- *none* – ҳаракатланиш якунлангач, дарҳол бошланғич ҳолатга ўтади.

- *forwards* – ҳаракат якунлангач охирги жойлашувда қолади.

- *backwards* – ҳаракатланиш якунлангач, бошланғич ҳолатга қайтади.

- *both* – ҳаракат якунланган ҳолатда қотади, *forwards* ва *backwards* қийматларидағи ҳолатлар нисбий бўлиб, улар «*animation-direction*» ва «*animation-iteration-count*» хусусиятларида кўрсатилган қийматларга боғлиқ. Масалан, йўналишда тескари қиймат берилган бўлса, охирги жойлашув тескари олинади.

▪ **Тўхтатиб туриш –**

animation-play-state: *paused* | *running*

- *paused* – ҳаракатланишни тўхтатиш

○ **running** — ҳаракатланишни давом эттириш
Сичқонча ёрдамида объект танланганда ёки босилганда ҳаракатланишни тўхтатиб, кейин керак бўлганда ўша жойидан давом эттириш учун бу усул кенг қўлланилади.

8.7.2. Ҳаракатланишни таъминлаш

Юқорида кўриб ўтган хусусиятларимиз, ҳаракатланиш қандай тарзда амалга ошишини белгилайди, аммо ҳаракатланишнинг ўзини таъминламайди. Бунда унинг «@keyframes» буйруғидан фойдаланилади.

```
@keyframes ном {  
    from { қоидалар }  
    to   { қоидалар }  
}
```

Бунда «ном» «animation-name»да кўрсатилган идентификатор, «from» дан кейин бошланғич ҳолатни аниқловчи CSS қоидалар, «to»дан кейин охирги ҳолат кўрсатилади. Бошланғич ва охирги ҳолатни ифодалаш учун «from» ва «to» сўзларининг ўрнига мос равишда «0%» ва «100%» сифатида кўрсатиш мумкин. Шундай экан, «animation-duration»да кўрсатилган вақтга нисбатан ихтиёрий фоиз хисобида, ўша пайтдаги вазиятни қоидаларини кўрсатиш мумкин. Агар «0%» кўрсатилмаса, унда ҳаракат дарҳол бошланмайди.

Мисол тариқасида 100px ўлчамдаги квадратни аввал энини тўрт марта ошириб, кейин бўйини икки марта узайтирадиган ҳаракатни чексиз тақрорланишини кўрамиз.

```
<div class="quiti"></div>  
  
.quiti {  
    background-color: green;  
    animation-name: kengay;  
    animation-duration: 2s;  
    animation-direction: alternate;  
    animation-iteration-count: infinite;  
}
```

```
@keyframes kengay {  
    0% { width: 100px; height: 100px; }  
    50% { width: 400px; height: 100px; }  
    100% { width: 400px; height: 200px; }  
}
```

Ҳар бир «from», «to» ёки фоизда кўрсатилган ҳолат ёзиладиган қоидалар ичida бир хилда хусусиятлар кўрсатиш тавсия этилади, акс ҳолда уни ҳаракатлантириш вақтида бошлангич қийматларига қараб яратилади ва кутилган натижা акслантирилмаслиги мумкин.

8.7.3. Хусусиятларни умумлаштириш

CSSда кенгайтирилган хусусият қийматларини ҳар бирини алоҳида бериш ўрнига уни умумий номланища бирлаштириш имконияти мавжуд. Яъни қуидагилар бир хил маънони англаради:

```
animation-name: harakat;  
animation-delay: 800ms;  
animation-direction: alternate;  
animation-iteration-count: infinite;  
animation-timing-function: ease-in;  
  
/* хамо умумлаштириши */  
animation: harakat 800ms alternate infinite ease-in  
/* тўғри умумлаштириши */  
animation: harakat ease-in 800ms infinite alternate
```

Қийматларни бўш жой билан ажратиб умумлаштириб берилаётганда уларнинг кетма-кетлиги ушбу мавзуда тавсифланган бўйича келиши талаб этилади. Агар бирор ҳусусият учун одатий қиймат қаноатлантирса, уни кўрсатмасдан тушириб қолдириш мумкин. Агар умумлаштирилган қийматларда кетма-кетлик ўзгарса, браузер берилган микдорнинг ёзувига қараб мос ҳусусиятга бириктиришга уринади, аммо эътибор бериш лозимки ушбу ҳолатда «animation-delay» ва «animation-duration»лар бир хил турдаги вақт қийматини қабул қиласди. Демак,

мисолда көлтирилган «800ms» вақт «animation-delay»га әмас, «animation-duration»га берилади.

Албатта умумлаштирилган қиймат бериш усули ёзувни қысқартириш билан бирга, код ёзилған файл ҳажмини камайтиради. Бу интернет трафик ҳажмини кам бўлишига олиб келади. Маълумот тез юкланиши билан бирга уни қайта ишлаш жараёни ҳам анча тезлашади. Аммо у камчиликдан ҳоли әмас. Масалан «animation-name»да бир неча ном кўрсатиш лозим бўлса, унга мос бошқа хусусиятларга ҳам шарт бўлмаса-да бир неча қиймат вергул билан ажратилган ҳолда берилиши керак бўлади. Умумлаштирилган қийматлардаги ёзувлар қўпайган сари уни мос хусусиятларга бўлиб тақсимлашда чалкашликлар юзага келади. Шунинг учун битта хусусиятда бир неча қиймат бериладиган ҳолда қийматларни умумлаштириш тавсия этилмайди.

Қуйида хусусиятларни ёритилаётганда, уларни умумлаштиришда бериладиган қийматлар кетма-кетлиги муҳим бўлса, қай тарзда көлтириш лозимлиги алоҳида баён этилган.

8.7.4. Содда ҳаракатлантиришлар

Аслида табақалаш [8.6.] қўлланилган обьектлар, маълум кўрсатилган ҳолат учун бошқа қоидалар берилган бўлса, ўша ҳодиса бажарилган пайтда дарҳол ўзгаришлар амалга оширилиши орқали, кескин ҳаракатланиш бажарилади. Масалан, обьектнинг устига кўрсаттич көлтирилганда (:hover) ёки босиб туриш орқали фаоллаштирилганда (:active) янги қоидаларга дарҳол ўтиш ўрнига ҳаракатлантириш ҳам жойлаш ҳам мумкин. Бундай холларда доим «@keyframes»ни қўллаш малол келиши табиий. Хусусиятлар қийматини ўзгартириш орқали пайдо қилинадиган бундай ҳаракатланишни ифодалаш учун «animation» ўрнига унинг соддалаштирилган талқини — «transition»дан фойдаланилади. У қуйидаги кенгайтирилган хусусиятларни ўз ичига олади:

- **Хусусият қийматини ўзгартириши:**
transition-property: none | all | хусусият

◦ none — ҳеч бир хусусият қиймати бўйича ўзгартирилмайди.

◦ all — барча ёзилган хусусиятлар ўзгаради.

◦ хусусият номли қийматигина ўзгаради. Агар бир неча хусусият бўйича ўзгариши талаб этилса, улар вергул билан ажратиб ёзилади.

▪ **Самарали ҳаракатланиш вақти:**

transition-duration: *вақт*

Самарали қўшимча ҳаракат қанча вақт давомида бўлишини кўрсатади, одатий қиймати 0, яъни ҳаракатланиш кўринмайди ва фақат кўчиш бўлади.

▪ **Ўтиш вақти тезлиги:**

transition-timing-function: *қиймат*

Бу самарали ҳаракатланишнинг содир бўлиш вақти ўзгариши, «*animation-timing-function*» сингари аниқланади. [8.7.1.]

▪ **Кутиш вақти:** **transition-delay:** *вақт*

Самарали ҳаракат бошланишдан аввал қанча кутиш лозимлигини кўрсатади. Вергул билан ажратиб бир неча қиймат бериш мумкин, уларнинг ҳар бири мос равишда «*transition-property*»да кўрсатилган хусусиятлар учун тадбиқ этилади. Мисол:

```
#tepa {  
    background-color: green;  
    height: 20px;  
    transition: height 2s ease-out 200ms;  
}  
#tepa:hover { height: 120px; }
```

8.7.5. Ҳаракатлантириш самаралари

Ҳаракатлантиришда шунчаки бошланғич ҳолатдан охиргисига маълум вақтда текис силжишилар орқали амалга оширмасдан, орада қўшимча ўзгаришлар, яъни самаралар қўшиш ҳам мумкин. Бунинг учун кўринишини ўзгартирувчи функциялардан фойдаланилади. Аввал қайси нуктага нисбатан ўзгариш амалга оширилишини аниқлаб олиш лозим, бунинг учун қуйидаги хусусият қўлланилади: **transform-origin:** *x y z*

Бунда Декарт координаталар тизимиға мутаносиб мос ўқлар бўйича қиймат киритилади. Одатий қиймати: 50% 50% 0. Жойлашишни англатувчи «left», «center», «right», «top», «bottom» сўзларини «x» ва «y» қийматларининг ўрнига қўллаш мумкин.

Ички объектлар уч ўлчамли фазода қандай аксланирилишини «transform-style» хусусиятида кўрсатилади. Одатий қиймати: flat, яъни ташқи элемент билан битта текислиқда. Иккинчи қабул қилиши мумкин бўлган қиймат: «preserve-3d». Уч ўлчамли шаклнинг фазовий айлантирища унинг орқа тарафини кўрсатишни «backface-visibility» хусусияти аниқлайди. У «visible» (орқа тарафи ҳам кўринарли, одатий қиймат) ва «hidden» (орқаси кўринмай, олди билан бир хил аксланади) қийматларидан бирини қабул қиласди. Албатта бу хусусиятнинг таъсири икки ўлчамли элементлар учун сезиларли бўлмайди.

Кўринишни ўзгартирувчи функциялар «transform» хусусиятига қиймат сифатида берилади. Бир неча функция кўрсатилиши керак бўлса, бўш жой билан ажратилади.

Объектни турган жойидан силжитиш учун қуйидаги функцияларидан мосини қўллаш мумкин: translate(x,y); translateX(x); translateY(y); translateZ(z); translate3d(x,y,z);

```
<div class="suril"></div>
.suril { transition-duration: 2s; }
.suril:hover { transform:translate(120px,20px); }
```

Объектнинг масштабини мос йўналишлар бўйича ўзгартириш учун қуйидаги функциялар ишлатилади: scale(x,y); scaleX(x); scaleY(y); scaleZ(z); scale3d(x,y,z); Мисол:

```
<div class="kattalash"></div>
.kattalash { transition-duration: 2.3s; .}
.kattalash:hover { transform: scale(4,3); }
```

Масштабни ўзгартириш учун олдин zoom хусусијати ишлатилган, унинг камчилиги шундаки, у факат битта ўлчам асосида (эни ва бўйи алоҳида берилмайди) ўзгартиради ва намойиш этилаётганда жойлашган

бошлангич нүктасининг координатаси сақланиб қолади. Бу эса ёнидаги обьектларнинг суримишига олиб келади. Уни вазифасини тўлақонли бажара оладиган «transform» қабул қиласидиган «scale» функцияси эса ҳажми ўзгараётган обьектнинг марказий нүктасига нисбатан олади ва бошқа элементлар жойлашишига таъсир ўтказмайди. Шу билан бирга «zoom»ни қўлламайдиган «FireFox»да ҳам ишлайди.

Объектни кўрсатилган ўққа нисбатан «a» бурчакка буриш учун `rotate(a);` `rotateX(a);` `rotateY(a);` `rotateZ(a);` `rotate3d(x,y,z,a);` функциялари, тик ёки ётиқ равишда маълум бурчакка оғдириш учун эса `skewX(a),` `skewY(a)` ва `skew(ax, ay)` функциялар ишлатилади.

Булардан ташқари тасвир нүқталарини матрица ёрдамида қайта аниқлаб акслантирувчи функциялар ҳам қўллаш мумкин. [6.4.3.]

Юқорида ўргангандан билимларимизни тадбиқ этган ҳолда 200px ўлчамли квадрат майдонни эгаллаган олма суратини курсор устига борган вақтда бир думалатиб бўёқли тасвир ҳолатига келтириб қўядиган CSS ни ёзиб кўрамиз:

```

.dumala { transition-duration: 2.7s; }
.dumala:hover {
    filter: invert(80%);
    transform: translate(200px, 10px) rotate(1turn);
}
```

8.8. Қаторли ва блокли теглар

HTML тилида теглар уч хил турга бўлинади. Биринчиси — **тавсиф теглари**, маълумот учун қўлланилади. Улар браузерда намоён бўлмайди, акслантирилайдиган теглар қандай қонуниятлар асосида пайдо бўлиши лозимлигини аниқлайди. Уларга «head» ва унинг ичидағи теглардан ташқари қуйидагилар ҳам киради:

**
** — Янги қаторга ўтишни таъминлаш;
<wbr> — Матн объект ичига сифмаган ҳолда кейинги қаторга тушишини таъминлаш;

Иккинчиси — блокли теглар, улар билан қўйида танишамиз. Учинчиси эса — қаторли теглар, матннинг айрим қисмларини алоҳида жило билан кўрсатиш учун мўлжалланган. Агар матн ичидаги икки сўзни ажратиб кўрсатиш лозим бўлса, уларни мос қаторли тегга олинганда, блокли тегдан фарқли ўлароқ, сўзларнинг жойлашишига қараб биттаси юқори, иккинчиси кейинги қаторга тушиб қолиши мумкин. Қаторли тегдаги биринчи сатрнинг қўйи қисми — «**асос чизиги**» деб юритилади, у хуснихат ўлчами ва номига қараб турлича баландликдаги бўлиши мумкин. Қаторли тегларнинг эни ичидаги элементнинг ҳажмита мос кенглиқдаги жойни эгаллайди, улар ичма-ич кўлланилиши мумкин. Қаторли тегларга қўйидагилар киради:

- <a>** — Ишорат жойлаштириш.
- <abbr>** — Қисқартма эканлигини англашиб.
- ** — Ҳарфларни қалинлаштириш.
- <bdi>** — Ёзув йўналишини ўзgartириш.
- <bdo>** — Матн йўналишини кўрсатиш.
- <cite>** — Кўчирма гап (бошқа манбадан олинган) эканлигини англашиб.
- <code>** — Дастурий код сифатида ёзилганлигини англашиб.
- ** — Ўчирилган ёзувни англашиб, устидан чизилган ҳарфлар билан ёзилади.
- <dfn>** — Атама эканлигини кўрсатиш, эгри тарзда ёзилади.
- ** — Ургу берилган матн бўлаги, эгри кўринишда акслантирилади.
- ** — Хуснихатни ўзgartириш (ишлатиш тавсия этилмайди).
- <i>** — Эгри ёзув.
- ** — Расм қўйиш.
- <input>** — Форма дарчасини қўйиш

- <ins>** — Кўшилган матн қисмини англатиш, тагига чизилган ҳолда ёзилади.
- <kbd>** — Киритиладиган матнлигини англатиш, кенг ҳарфлар билан ёзилади.
- <label>** — Форма элементига номланиш қўяди.
- <mark>** — Матннинг муҳим қисмини англатади.
- <meter>** — Оралиқдаги қийматни тасвирлаш. [7.4.6.]
- <q>** — Қисқа қўчирмани англатади.
- <rp>** — <ruby> тегини кўлламайдиган браузерларда нима чиқишини кўрсатади.
- <rt>** — <ruby> объектининг қуий ёки тепасида нима чиқишини кўрсатади.
- <s>** — Устидан чизилган ёзув.
- <samp>** — Дастур натижаси сифатида чиқсан ёзувни англатади.
- <small>** — Кичик ўлчамдаги ёзув
- <select>** — Тайинланувчан дарча (очилувчи рўйхат) қўйиши. [7.4.3.]
- ** — Оддий қаторли тег яратиш.
- ** — Муҳим матннинг қисмилигини англатиш, қалин ҳарфлар.
- <sub>** — Қуий индексдаги ёзув, масалан: H_2O
- <sup>** — Юқорироқда даража сифатидаги ёзув, масалан: x^2
- <textarea>** — Матн киритиш дарчаси.
- <time>** — Сана ва вақтни англатувчи ёзув.
- <var>** — Ўзгарувчи эканлигини англатиш, эгри ёзувда ифодаланади.
- <video>** — Лавҳа файл қўшиш.

Блокли теглар доим тўғри тўртбурчак шаклидаги майдонни эгаллайди. Одатда улар алоҳида қатордан ўрин олиб, бутун кентликни эгаллайди. Қаторли теглар сингари матн ичидаги сўзлар орасида жойлашиб кетавермайди. Қаторли теглар ичига блокли теглар жойлаш мумкин эмас! Умуман олганда, қаторли ва блокли тегларнинг бошқа ажralиб турадиган хусусиятлари қолмаган. CSS ёрдамида қаторли тегни блокли тегга ва аксинча айлантириш имконияти мавжуд. Бунинг учун **display**

хусусиятидан фойдаланилади. қиймати аниқланган тур:

- **none** — яширин, объект акслантирилмайды.
- **inline** — қаторли тег объекти сифатида аксланади.
- **block** — блокли тег объекти сифатида аксланади.

○ **inline-block** — ўзи қаторли тег кўринишида ва ички қисми блокли тег сифатида намоён бўлади. Яъни, олдинги элемент якунланган жойдан бошланса ҳам, унинг эни ва бўйини ўзгартириш мумкин.

○ **inline-table** — блокли тег сифатида аксланади, аммо атрофини бошқа элементлар, хусусан матн билан ўраш мумкин бўлади.

○ **flex** — блокли тег бўлади, фақат ичидаги қўшни теглар бир нечта бўлса, юзаси шу объект ўлчамига қараб бир хилда ўзгариб ('«эгилувчан» [9.1.]) жойлашади.

○ **inline-flex** — «flex» хусусиятли ('«эгилувчан жойлашув» бўйича) қаторли тег.

○ **list-item** — рўйхатдаги сингари нишончага эга блокли тег бўлади. [3.4.2.]

○ **run-in** — қаторли ёки блокли сифатида қатъий аниқланмаган тегларнинг ичидаги ташкил этувчилирига қараб ўзи аниқлаш имконини беради. Буни қўллаш тавсия этилмайди, баъзи браузерлар тўғри акслантири маслиги мумкин.

○ **table** — жадвал объекти сингари намоён бўлади.

○ **table-caption** — жадвал сарлавҳаси хусусиятлари билан акслантирилади.

○ **table-cell** — жадвал катаги шаклида, «td» ёки «th» хусусиятларини олади.

○ **table-column** — жадвалдаги «col» теги хусусиятлари билан аксланади.

○ **table-column-group** — «colgroup» объекти хусусиятлари билан намоён бўлади.

○ **table-footer-group** — «tfoot» сингари охирги бир неча сатрни акслантиради.

○ **table-header-group** — «thead» сингари бир неча бошлиғийч қаторни сақлайди.

○ **table-row** — жадвал қатори («tr») сифатида намоён бўлади.

○ **table-row-group** – «tbody» сингари бир неча қаторни кўрсатувчи блокли тег.

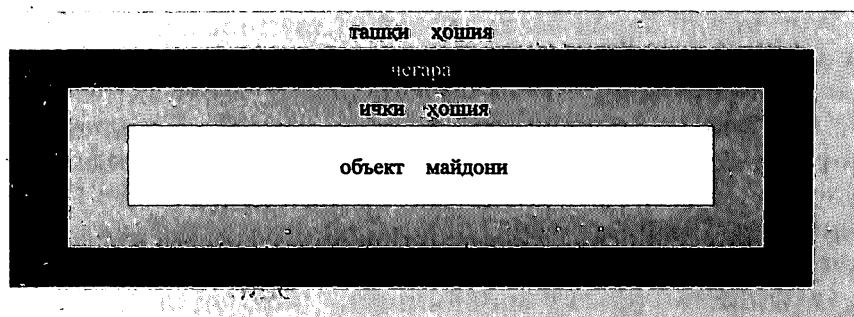
Қуийдаги теглар блокли ҳисобланади:

- | | |
|---|---|
| <address> | — Муаллиф ёки ташкилот манзилларини кўрсатади. |
| <article> | — Матнни мақола, янгилик ёки маҳсус хабарлигини англатади. |
| <aside> | — Саҳифанинг ён қисмидаги бўлакни билдиради. |
| <audio> | — Товуш файлли жойлаштириш. |
| <button> | — Тугма яратиш. |
| <blockquote> | — Бошқа манъбадаги маълумотни ажратиб кўрсатади. |
| <canvas> | — Тасвир чизишни таъминлаш. |
| <div> | — Саҳифанинг алоҳида бўлими. |
| <dl>,<dd>,<dt> | — Изоҳли рўйхат тузади. [3.4.3] |
| <figure> | — Саҳифадаги элементларни гурӯҳлайди. |
| <footer> | — Саҳифанинг қуий қисми. |
| <form> | — Форма яратиш. |
| <h1> | — Биринчи даражали сарлавҳа. |
| <h2> | — Иккинчи даражали сарлавҳа. |
| <h3> | — Учинчи даражали сарлавҳа. |
| <h4> | — Тўртингчи даражали сарлавҳа. |
| <h5> | — Бешинчи даражали сарлавҳа. |
| <h6> | — Олтинччи даражали сарлавҳа. |
| <header> | — Саҳифа ёки бўлимнинг бош қисмини аниқлайди. |
| <hr> | — Уфқий чизиқ қўяди. |
| <iframe> | — Фрейм қўйиши. |
| <nav> | — Ишоратлар гурӯҳини аниқлайди. |
| , | — Саноқли рўйхат тузади. [3.4.1.] |
| , | — Белгиланган рўйхат тузади. [3.4.2.] |
| <p> | — Хатбошини аниқлайди. |
| <pre> | — Таркибини ёзувдаги белгиланишлар сақланган ҳолда кўрсатади. |
| <ruby> | — Катта бўлмаган тавсифномалар, кириш сўзларини белгилайди. |

- <section>
<table>
- Саҳифа бўлимини анлатади.
 - Жадвал яратади. [4.4.]

8.9. Объект шакли

Блокли тег тўғри тўртбурчак шаклида намоён бўлади. Унинг чегаравий чизиги, яъни периметри ҳамда ички ва ташқи ҳошияси мавжуд. Ташқи ҳошия — чегаравий чизикдан кўшни объектларгача ёки ўз ташқи тегининг майдони қиррасигача бўлган оралиқ. Ички ҳошия — бу чегаравий чизикдан ўз элементлари қанчалик ичкарида жойлашишини аниқловчи масофа.



Куйидаги хусусиятлар билан объект ўлчамини аниқланади:

- **padding** — ички ҳошия қалинлиги, қийматини «ўлчам» ёки фоиз [2.6.] ҳисобида «соат мили» қоидаси асосида берилади. Яъни, қиймат биттадан тўрттагача сон билан аниқланади. Агар битта сон берилган бўлса, тўрт тарафдаги ички ҳошия бир хилда ўша сонга teng бўлади. Йккита сон бўлса, юқори ва пастги ҳошиялар биринчи сонга, ён ҳошиялар иккинчи сонга teng бўлади. Учта сон билан берилса: биринчиси юқорини, иккинчиси ён томонларини, учинчиси пастки ҳошияни анлатади. Тўртта сон бўлса, улар мос равишда юқори, ўнг, паст ва ҷаҳ ҳошияларни билдиради. Одатий қиймати: 0. Уларни ҳар бирини алоҳида «padding-top», «padding-right», «padding-bottom» ва «padding-left» кенгайтирилган хусусиятлар ёрдамида ҳам ифодалаш мумкин.

▪ **margin** — ташқи ҳошия қалинлигини ўлчамларда кўрсатувчи умумлаштирилган хусусият. У «**margin-top**», «**margin-left**», «**margin-bottom**» ва «**margin-right**» кенгайтирилган хусусиятларини «соат мили» қонунияти асосида аниқлайди.

▪ **visibility** — Объектнинг қай тарзда кўринишини таъминлайди, қиймати:

◦ **visible** — одатий қиймат, ўз ҳолиша кўринади;

◦ **hidden** — яширин, бунда барча эгаллаб турган жойи, яъни ўлчамлари сақланиб қолади. Фақат объект кўринмайди, яъни шаффоф рангда намойиш этилади.

◦ **collapse** — бу фақат жадвал катаклари учун қўлланилади, агар ягона катакка тадбиқ этилса, шунчаки кўринмас ҳолда жой эгаллайди, агар қатор ёки устуннинг барча катакларига қўлланилса, у жадвалдан ўчирилади. Матн учун қўлланилганда «**hidden**» сифатида ишлайди.

▪ **height** — объект майдонининг баландлиги, унга чегара ва ҳошияларнинг қалинлиги кирмайди. Қийматини ўлчам ёки фоиз ҳисобида бериш мумкин. Одатий қиймати: «**auto**», яъни ички ташкил этувчилирининг умумий баландлигидан келиб чиқиб белгиланади.

▪ **width** — объект майдонининг эни, унда ҳам чегара ва ҳошиялар қалиниклари ҳисобга олинмайди. 100% қиймат берилганда, у ташқи объект кенглигини тўла эгаллайди. Айтайлик, унда «**padding: 10px; margin: 8px;**» деб кўрсатилган бўлса, бу ҳолда элемент ташқи объектга сиғмай айлантиргич пайдо бўлади ёки бир қисми қирқилиб кўринмай қолади. Сабаби иккита ён томонларини ҳисобга олганда объектнинг эни 100% дан «**36px**»га катталашиб қолди. Бундай вазиятда, айлантиргич пайдо қилмасдан объектни тўла кўрсатиш учун унинг энига қўйидагича қиймат берилади: [8.4.10.]

```
width: calc(100% - 36px);
```

▪ **box-sizing** — объектнинг эни ва бўйи ўлчамини ҳисоблаш алгоритмини аниқлайди. Қиймати аниқланган тур:

◦ **content-box**: одатий қиймат, бунда «**width**» ва

«height»да күрсатылған қийматта ички ва ташқи ҳошиялар ҳамда чегара қалинлиги кирмайды.

◦ **border-box**: күрсатылған қиймат ўз ичига ички ҳошия (padding) ва чегара қалинлигини (border-width) олади, ташқи ҳошия (margin) бунга кирмайды.

▪ **box-shadow** – Объектта соя құшиб, унинг умумий ўлчамини оширади. Одатай қиймати: «none», лекин қиймат бериш синтаксиси бироз мураккаброқ:

box-shadow: [inset] x y таралиш чўзилиш ранг;

Бунда «inset» сояни объектнинг ичида бўлишини таъминлайди, агар күрсатылмаса ташқарига тушади, «x» ва «y» мос ўқ бўйича сурилиш ўлчови. Соя узокроқقا ёрқин рангда тарқаши учун «таралиш»ига каттароқ мусбат қиймат бериш зарур, одатда унинг қиймати 0, у битта рангда аниқ чегарада бўлишини таъминлайди. Соя қалинлигини «чўзилиш»га берилган қиймат аниқлайди, одатай қиймати 0 ва бу ҳолда соя объектнинг ўлчамига тенг ҳисобланади. Соя «ранг»и күрсатылмаса, у қора сифатида олинади. Агар соя бир неча қатламдан иборат бўлиши керак бўлса, қийматлар вергул билан ажратилиб шу тартибдан бир нечтасини ёзиш мумкин.

Турли кенгайтмали мониторларда ва браузер ҳажми ўзгартирилганда аниқ ўлчами берилмаган теглар учун акслантирилаётган объектларнинг юзаси ўзгаради. Бунинг натижасида мухим маълумот сигмай қолиши ёки сурилиб кетганилиги боис керакли жойда кўринмаслиги мумкин. Ўлчам ўзгарап экан, бежиirimлигини йўқотмаслик учун объектларнинг энг катта ва энг кичик қабул қилиши мумкин бўлган қийматларини қатъий белгилаб кўйиш даркор бўлади. Бундай вазифани мос равища «max-height», «max-width», «min-height» ва «min-width» хусусиятлари бажаради.

Чегаравий чизиқнинг қалинлигини «border-width» хусусияти ёрдамида берилади. Қийматига «ўлчам» ёки **thin** (2px), **medium** (4px) ва **thick** (6px) сўзларидан бирини бериш мумкин. Одатай қиймати: «medium».

Ҳошиялардан фарқли тарзда чегара маҳсус чизиқ ёр-

дамида акслантирилади. Агар у кўринувчан бўлса, рангини ҳам ўзгариши имкони мавжуд бўлиши керак. Бунинг учун «**border-color**» хусусиятидан фойдаланилади. Қиймати: «ранг». Агар кўринмас бўлиши лозим бўлса, шаффоф ранг берилади, у «**transparent**» сўзи билан ифодаланилади.

Чегаранинг қандай тарзда чизилишини «**border-style**» хусусияти билан аниқланади. Қиймати чизик шаклини кўрсатувчи аниқланган тур:

- **none** — одатий қиймат, чизик кўринмайди, қалинлигини 0 қилиб қўяди.
- **hidden** — кўринмас;
- **dotted** — нуқтали;
- **dashed** — (узук) чизиқчали;
- **solid** — узлуксиз чизиқли;
- **double** — иккита қават чизиқли;
- **groove** — ботиқ чизиқли;
- **ridge** — бўртиқ чизиқли;
- **inset** — ичкарига соя тушгансимон;
- **outset** — ташқарига соя тушгансимон;

Чизикнинг қалинлигини, рангини ва шаклини билдирувчи бу учала хусусият ҳам «соат мили» қоидаси асосида тўртала тарафга турлича тадбиқ этилиши мумкин. Қийматлари тўртала томон учун бир ҳил бўлганда, яъни биттадан кўрсатиладиган ҳолда уларнинг ягона «**border**» хусусиятига бирлаштириш мумкин. Бунда қайси кенгайтирилган хусусият қиймати қаерда келишининг аҳамияти йўқ. Ранг, ўлчам ва аниқланган турни англатувчи турлича қийматлардан браузернинг ўзи мосини керакли хусусиятга бириттириб олаверади.

```
border: 2px inset green;
```

Бунда тўрттало томонга ҳам бир ҳил қоида тадбиқ этилади. Аниқ бир томонни бошқача тарзда чизиш лозим бўлса, унинг учун алоҳида хусусият ишлатилади: «**border-top**», «**border-bottom**», «**border-left**», «**border-right**». Булар ҳам мос учта кенгайтирилган хусусиятни умумлаштиради. Агар айни битта томонга яона кенгайтирилган хусусият қиймати берилиши шарт бўлса,

танишиб чиқылған қонунияттар асосида қуйидаги барыш мүмкін:

<code>border-top-color</code>	<code>border-top-style</code>	<code>border-top-width</code>
<code>border-right-color</code>	<code>border-right-style</code>	<code>border-right-width</code>
<code>border-bottom-color</code>	<code>border-bottom-style</code>	<code>border-bottom-width</code>
<code>border-left-color</code>	<code>border-left-style</code>	<code>border-left-width</code>

Мисол тариқасида тағига узук чизикли чегара қўйиладиган объектни кўрамиз:

`border-bottom-style: dashed;`

Жадвалдаги катакларни ёки ёнма-ён турувчи блокли тег объектлари чегараловчи чизиқлар алоҳида-алоҳида бўлиб, икки қатламли чизиқ сифатида ифодаланади. Бу эса хунук кўриниш касб этади. Жадвалда чизиқларни битталик қилиб чизиш учун «cellspacing» аломатига 0 қиймат берилади. Бунинг CSSдаги муқобили «border-collapse» хусусиятидир. Қиймати аниқланган тур:

- `collapse` — чегарани англатувчи чизиқлар ёнма-ён катаклар учун бирлаштириб, ягонадай ифодаланади. Бунда жадвалдаги «cellspacing» аломат қиймати инобатга олинмайди.

- `separate` — чегаравий чизиқлар ҳар бир объект учун алоҳида чизилади. Ёнма-ён объектлар учун икки қатламчи чизиқ намоён бўлади.

Ушбу хусусият жадваллар ёки «display» хусусиятида жадвалга оид қиймат берилган объектлар учун ишлайди. Жадвалнинг яна бир аломати «cellpadding» [4.1.] вазифасини CSSдаги «border-spacing» хусусияти бажаради.

Тўғри тўртбурчак шаклидаги чегара қирраларининг тўғри бурчак шаклида бўлиши замонавий талабларга жавоб бермайди. Бежирим кўриниши учун қирраларнинг ёйсимон қилиб қўйиш одат тусига кириб улгурди. Авваллари бунинг учун тўртта четта маҳсус расм қўйиб, уни чегаравий чизикка бирлаштириб юбориш машаққати мавжуд эди. Лекин чиройли кўри-

нишни таъминлаш учун дастурловчи дизайннерлар анчагина ортиқча мәхнатни амалга оширишга ҳам мажбур бўлишган. Ҳозирда қиррани ўтмаслаштирувчи «**border-radius**» хусусияти мавжуд бўлиб, у барча замонавий браузерларда кўлланади. Унга соат мили қоидаси бўйича берилган ўлчамдаги радиусли айлана ёйи чегаравий чизикларга бирлаштирилиб, кирра ўрнига ёй чизилади. Агар ёй айлана эмас, эллипснинг қисми сифатида кўриниши лозим бўлса, қийматда «/» дан сўнг иккинчи радиус берилади.

Айнан фақат битта бурчакни ёйсимон қилиш учун маҳсус «**border-top-left-radius**», «**border-top-right-radius**», «**border-bottom-left-radius**» ва «**border-bottom-right-radius**» хусусиятларидан бирини кўллаш кифоя. Таъкидлаб ўтиш лозимки, ушбу хусусият «**border-collapse: collapse**» қийматини олган жадваллар учун ишламайди.

Чегарани оддий чизиклар билан чизиш ҳар доим ҳам талабни тўла қондира олмайди. Баъзан маҳсус расм асосида шакл беришга тўғри келади. Лекин расмнинг ўлчамини олдиндан катталиги аниқ бўлмаган объектта мослаш бироз мураккаблик туғдиради. Бундай муаммони ҳал этиш учун «**border-image**» хусусияти қўлланилади. Унинг тадбиқ этиш алгоритми қуйидагича: аввал тўғри тўртбурчак шаклида безакли чегарага эга расм (рамка) танланади. Унинг қайси томонидан қанча нуқта бирлигига кесиб олиниши лозимлиги аниқланади. Кесилган тасмаларни объектнинг четига маълум масштаб асосида ўзгартириб, учларини тақаб қўйилади, ортиб ёки етмай қолган томонлар асосига тасма бўлаклари чўзиб, такрорланиб ёки кесилиб тўлдирилади. Шу тариқа расмдаги ҳошиявий безакнинг объектта мос масштабланган кўриниши пайдо бўлади. Ушбу хусусиятнинг умумий синтаксиси қуйидагича:

border-image: расм энц [/қалинлиги] [жойлашиши];

Бунда url функцияси ёрдамида «расм»нинг манзили кўрсатилади, расмнинг қайси томонларидан қанча нуқтадан қирқиб олиш лозимлигини «энц»да, уни

қанчага күпайтириб қўйиш лозимлиги «қалинлигид»
кўрсатилади. Бу икки қиймат «соат мили» қоидаси
асосида биттадан тўрттагача сон қабул қилиши мум-
кин. Фақат уларда бутун сон кўрсатилади, «ўлчам» эмас
(10px сифатида бериш хато). Мос томон учун расмдан
кесиб олинган тасмани қандай қўйиш лозимлигини
«жойлашиши»да кўрсатилади. Унинг одатий қиймати:
«stretch» бўлиб, унда тасмани чўзиб ёки сиқиб кўрса-
тилган ўлчамга мосланади; «gereat» қиймати берилса,
қирғоқقا расм ўз ўлчамида такрорий давом эттирилади;
«round»ни қабул қилганда тасма шундай масштабда ўз-
гартириб жойланадики, натижада такрорланадиган расм
бутунилигини сақлайди.

```
border-image: url(hoshiya.png) 30 20 / 1 1.5 round round;
```

Таъкидлаб ўтиш лозимки, ушбу хусусият «border-collapse: collapse» қийматини олган обьектлар учун иш-
ламайди.

Шаклни ифодаловчи юқоридаги хусусиятларни, ўл-
чамни аниқловчиларини истисно қилган ҳолда, қаторли
теглар учун ҳам тадбиқ этиш мумкин.

IX БЎЛИМ

ОБЪЕКТЛАРНИ ЖОЙЛАШ ВА ЖИЛОЛАШ



9.1. Объектни жойлаштириш

Объектларни жойлаштириш ва жилолаш учун фақат CSS имкониятларидан фойдаланиш тавсия этилади. Юқорида таъкидлаганимиздагидек, ҳар хил экранларда веб-саҳифа ўзига хос тарзда намоён бўлиши керак.

Объектни HTMLда қандай эълон қилинишидан қатъий назар, уни CSS ёрдамида нафақат кўринишини, балки жойлашадиган ўрнини ҳам ўзгартириш мумкин. Жойлашишни аниқловчи хусусиятлар одатда блокли тегларга тадбиқ этилади, лекин жуда зарур бўлса, қаторли тегни «display» ёрдамида турини ўзгартириб, унга ҳам қўллаш мумкин. Қўйида жойлашишни белгиловчи муҳим хусусиятлар билан танишамиз.

Элементнинг тепадан, ўнгдан, пастдан, чапдан ўз ташқи объектига нисбатан қанча масофада жойлашишини мос равишда «`top`», «`right`», «`bottom`», «`left`» хусусиятлари билан берилади. Уларни келгусида «атроф хусусиятлари» деб юритамиз. Қийматини ўлчам ёки фоизда берилади. Одатий қиймати: «`auto`» ва у объект жойини ўзгартирмайди.

Браузер ойнаси ёки бошқа объектларга нисбатан қандай жойлашишини қўйидаги хусусият билан аниқланади:

```
position: absolute | fixed | relative | static | sticky;
```

Бунда:

- `absolute` — объектнинг ҳажми бошқа элементлар-

нинг жойлашишига таъсир қилмайди, гўё у ўзини бошқа параллел текислиқда (қатламда) тургандай тутади. Атроф хусусиятлари (left, top, right, bottom) браузер ойнаси бўйича, аниқроғи «body» аксланган майдонга нисбатан олинади.

○ **relative** — нисбий жойлашув, бунда атроф хусусиятлари ўзи турган ўрнига нисбатан қанча масофага кўчишини кўрсатади. Шу ўринда муҳим эслатма: агар «relative» қиймат берилган обьект ичига бошқа «absolute» қийматли элемент жойланса, унинг атроф хусусиятлари «body»га нисбатан эмас, ташқи обьект жойлашувига нисбатан олинади.

○ **fixed** — жойлашиши бўйича «absolute» қийматига ўхшайди, аммо айлантиргич ёрдамида саҳифа сурилганда ҳам у ўз ўрнини ўзгартирумайди. Бу қийматдан кўпинча реклама ойналарини қўйишда фойдаланилади.

○ **static** — обьект одатдагидек аксланади, бунда атроф хусусиятларининг қиймати таъсир қилмайди.

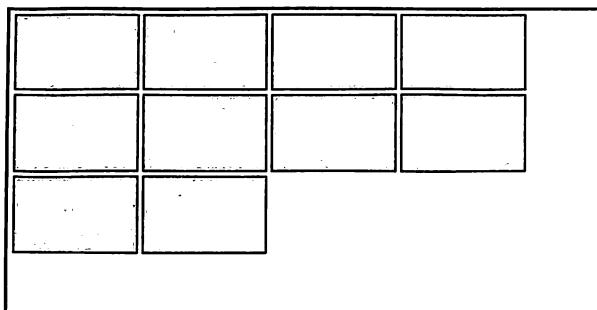
○ **sticky** — ўзгарувчан жойлашув. Маълум чегара доирасида ўзини кўчувчан тутади, чегара ташқарисида эса ўзгармас холатда қотади. Бу одатда суриладиган матнга эга бўлган саҳифаларда сарлавҳа учун ишлатилиди. Айлантиргич билан сурилганда матн кўринадиган вазиятда кўчиб боради, сурилаётган обьект экранда кўринмай қолганда, у ҳам жойини ўзгартирумасдан қотади.

Жойлашуви ва атроф хусусиятларининг баъзи қийматларида обьектнинг ўлчами мос равища ўзгаради, бунда «width» ва «height» қийматлари инобатта олинмаслиги мумкин. Масалан:

```
position: relative;  
top: 3px;  
right: 2px;  
bottom: 3px;  
left: 2px;
```

Шу ўринда «эгилувчан жойлашув» тушунчасига тўхвалиб ўтсан, бир нечта обьектларни биттасини ичида

қандай күринища намоён бўлишини аниқлашда бу усул кенг қўлланилади. Аслида у катта кути ичида кичик қутиларни тартиб билан тахлашга ўхшайди. Битта қатор тўлгандан кейин (ортиб қолган жойга битта кути сифмаган тақдирда), иккинчисига ўтилади, фақат охирги қатордаги қутилар сони умумий миқдорга қараб, одингиларидан кам бўлиши мумкин. Бунда қутичалар уфқий ва тиккасига текис катаклар сифатида жойлашади.



Ушбу расм «эгилувчан жойлашув»ни тўғри тасаввур қилиш учун имкон беради. Ташқи қутини «эгилувчан майдон» деб юритамиз. Бунда ҳар бир ички объектнинг атроф хусусиятлари бир хил қиймат қабул қиласди. Бундай усулда жойлашишни бошқариш учун қуидаги хусусиятлар ишлатилади:

- **flex-direction** — элементларнинг жойлашиш йўналишини аниқлайди. Қиймати:
 - **row**: жойлашиш тартиби матн ёзилиш йўналишида бўлади (одатий қиймат);
 - **row-reverse**: жойлашиш тартиби матн ёзилиш йўналишига тескари бўлади;
 - **column**: устун шаклида тепадан пастга йўналади;
 - **column-reverse**: устун шаклида пастдан тепага қараб жойлашади;
- **flex-wrap** — элементларнинг қаторда жойлашини аниқлайди. Қиймати:
 - **nowrap**: ички элементлар фақат битта қаторда жойлашади (одатий қиймат);
 - **wrap**: ички элементлар «flex-direction»да кўрсатил-

ган йўналишда бир неча қаторда жойлашишини таъминлайди.

◦ **wrap-reverse**: ички элементлар «flex-direction»да кўрсатилгандагига тескари йўналишда бир неча қаторда жойлашишини таъминлайди.

▪ **flex-flow** — умумлашган хусусият «flex-direction» ва «flex-wrap» қийматларини бирданига бериш имкониятини яратади.

▪ **flex-grow** — ички элементлар ўлчамини пропорционаллигини таъминлайди. Агар барча элементлар 1 қиймат қабул қилинган бўлса, уларнинг барчаси тенг ҳажмда бўлади, қайсиdir бирига 2 қиймат берилган бўлса, у бошқаларидан икки баробар катта намоён бўлади. Қиймати: сон.

▪ **flex-shrink** — сиқиши коэффиценти. Барча ички элементларни битта қаторда жойлаштириш учун обьектнинг бошқаларига нисбатан неча баробар қисқартириш мумкинлигини аниқлайди. Қиймати: бутун сон.

▪ **flex-basis** — ички элементларнинг бошланғич кенглигини аниқлайди. Одатий қиймати: «auto», аслида ўлчам ёки фоиз ҳисобида эни кўрсатилади.

▪ **flex** — «flex-grow», «flex-shrink» ва «flex-basis»ларнинг умумлаштирувчи хусусият.

▪ **order** — «эгилувчан» элементларнинг ёзилиш тартибида эмас, кўрсатилган рақамлар бўйича кетма-кетликда намоён бўлишини таъминлайди. Мисол:

```
.quti:nth-of-type(1) { order: 3; }
```

9.2. Орқа тасвир

Юқорида обьектнинг чегарасини тасвирлаш имкониятлари билан танишган эдик. [8.9.] Шунга ўхшаш хусусиятларни орқа тасвир учун ҳам қўллаш мумкин. Ранги қуийидаги хусусият билан қўйилади:

```
background-color: ранг | transparent;
```

Одатий қиймати: «transparent», шаффоф рангни англатади. Бунда күрсатилган ранг объектни түлиқ қоплайды ва ундағы элементлар орқасида тасвириланади. Агар ягона рангнинг ўрнига расм ёки градиент қўйилиши лозим бўлса, «background-image» [8.4.9.] хусусиятидан фойдаланилади. Унда бир нечта расмни вергул билан ажратиб кўрсатиш мумкин. Кўрсатилган расм юкландунча «background-color»да берилган ранг аксланиб туради. Ортрасм объект ташкил этувчилиари билан бирга сурилишини таъминлаш учун қуидаги хусусиятдан фойдаланилади:

background-attachment: fixed | scroll | local;

- **fixed** — ортрасм ҳаракатланмайди;
- **scroll** — ташкил этувчилиар билан бирга сурилади (одатий қиймат);
- **local** — агар объект айлантиргичга эга бўлса, сурилади. Фақат элементнинг ташқарисига чиқиб кетган ортрасм қисми ўз жойида қолади.

Чегарани узлуксиз чизиқ билан берилмайдиган ҳолларда, ортрасм чегаранинг айнан қайси қисмидан бошлаб кўриниши кераклигини белгилаш муҳим аҳамиятга эга. Чегарани ифодаловчи тасвирининг шаффоф қисмлари мавжуд бўлса, ортрасм ўша жойларда ҳам ўз ифодасини топиши мақсадга муроффақ бўлади. Бундай имкониятни аниқ белгилаш учун қуидаги хусусият қўлланилади:

background-clip: padding-box | border-box | content-box | text;

- **border-box** — одатий қиймат, ортрасм чегара тагида ҳам аксланади;
- **padding-box** — ортрасм чегара ичида (ички ҳошиядан бошлаб) аксланади;
- **content-box** — ортрасм фақат объект майдонининг ўзида аксланади.
- **text** — ортрасм фақат матн ортида аксланади.

Умуман олганда ушбу хусусият чегара ва ички ҳошия қалинлигига қараб ортрасмниң қаердан бошлиб акслантириш лозимлигини аниқлайди. Аслида айнан шунинг учун «background-origin» хусусияти ҳам киритилган, фақат бу «background-attachment: fixed;» қиймат берилганды инобатта олинмайди. Қолган ҳолларда қабул киладиган қийматлари ва вазифалари айнан «background-clip» билан бир хил. Фақат «background-clip: text;» қиймати бундан мустасно, ушбу қиймат ўзига хос бўлиб, эски браузерлар учун «-webkit-background-clip: text;» сингари ёзилиши талаб қилинади.

Расмнинг ўлчами ҳар доим ҳам объектни аксланаётган ҳажмига мос келавермайди. Бундай вазиятларда «background-size» хусусияти ёрдамида масштаблашга тўғри келади. Қиймат сифатида иккита сон: эни ва бўйини, ўлчам ёки фоиз ҳисобида бериш мумкин. Агар битта қиймат берилса, у эни ҳисобланаб, бўйи унга пропорционал равишда ҳисоблаб топилади. Аниқ ўлчамлар берилганды ортрасм кўрсатилгандай тўғри тўртбурчак шаклида намоён бўлади, фоиз кўрсатилганды объектнинг ҳажмига нисбатан ҳисобланади. Одатий қиймати «auto» берилса, ҳажми ўзгармасдан акслантирилади. Агар битта қиймат берилиб, иккинчиси сифатида «auto» ёзилса, иккинчиси пропорционаллик сақланган ҳолда биринчига нисбатан ҳисобланаб топилади. Қиймат сифатида «cover» ёзилса, ортрасмни тўлалигича намоён этади, бунда масштаб шундай ўзгарадики, эни ёки бўйи объект ўлчамига тўғри келади. Қиймат ўрнига «contain» берилганды ортрасмни масштаблаб, объектнинг ичида бўш жой қолмайдиган тарзда жойлаштирилади. Бунда ортрасмнинг эни ёки бўйидан бир бўлаги кўринмай қолиши мумкин.

Кўрсатилаётган расм ўлчами ўзгартирилар экан, катта ҳажмдаги файлни қайта ишлашда бироз вақт талаб қилиниши мумкин. Бундай вазиятларда масштаблаш услубини ўзгартириш орқали керакли натижага эришиш мумкин, «image-rendering» хусусияти қандай усулда масштаблаш кераклигини аниқлайди. Қиймати аниқланган тур:

- **auto** – браузернинг ўзидағи алгоритм бўйича ўзгартирилади, одатда у бикубик услугб хисобланади;
- **crisp-edges** – бунда тасвирдаги рангларнинг чегараловчи ва уларга яқин нуқталарни акслантириш билан тез намойиш бўлиши таъминланади;
- **pixelated** – катталаштирилаётганда тиниқлик сақланиши таъминланади, кичиклашаётганда «auto» сингари ишлайди. Бунда расм секин пайдо бўлади.

Ҳар доим ҳам ортрасм объектни тўлалигича қоплаши талаб этилмайди. Масалан тутгиларга нишонча қўйиш лозим бўлса, ундаги ном олдида кичик расм акслантирилиши керак. Расмнинг ўлчамига қараб, уни жойлашадиган ҳолатини ҳам ўзгартириш талаб этилади. Бунинг учун қуйидаги хусусиятидан фойдаланилади:

background-position: жойи;

Бунда «жойи» объектнинг чап юқори қиррасига нисбатан тик ва уфқий қандай масофада жойлашиши бўш жой билан ажратиб кўрсатилиади. [8.4.9.] Бундан ташқари фақат уфқий жойлашиши «background-position-x» ёки фақат тик жойлашиши «background-position-y» хусусиятлари билан кўрсатилиши мумкин.

Аксарият ҳолларда ортрасмни такрорлашга тўғри келади. Катта-катта объектларнинг ортини тасвирга тўлдириш учун аслида ҳажми кичик расмдач фойдаланиб, уни такрорий намойиш этилади. Такрорланишини бошқариш учун «background-repeat» хусусияти ишлатилиади. Ў қуйидаги қийматлардан бирини қабул қиласи:

- **repeat** – одатий қиймат; ортрасм ҳам уфқий, ҳам тик такрорланади.
- **no-repeat** – такрорланмайди, ортрасм кўрсатилган нуқтада чиқади.
- **repeat-x** – фақат уфқий такрорланади.
- **repeat-y** – фақат тиккасига такрорланади.
- **space** – бутун сондаги расмлар кўринадиган ҳолда такрорланади, бунда оралиқ масофалар ўзгартирилиши мумкин.
- **round** – бутун сондаги расмлар кўринадиган

холда такрорланади, бунда ортрасынан масштаби ўзгариши мүмкин.

Қийматларни уфқий ва тик равищда алоҳида-алоҳида «repeat-x» ва «repeat-y» билан бериш ўрнига бўш жой билан ажратиб кетма-кет бериш мүмкин. Масалан:

```
background-repeat: space round;
```

Бештагача ортасвирини аниқловчи кенгайтирилган хусусиятларни ягона тарзда «background»га умумлаштириш мүмкин. Бунда кетма-кетлик муҳим аҳамиятга эга эмас. Бунда «background-size» қиймати «background-position»дан ажратиш учун «/» билан ажратиб ёзилади. Одатий қиймати:

```
background: transparent none repeat scroll 0% 0%;
```

Аслида «background-color», «background-image», «background-repeat», «background-attachment» ва «background-position»ларнинг одатий қийматларини умумлаштирган.

9.3. Ташқи элементларга нисбатан жойлаш

Объектнинг ташқи ва қўшни элементларга нисбатан қандай жойлашишини қўйидаги хусусиятлар ёрдамида кўрсатилади:

- **float** — объекти қайси томонга текисланishi лозимлигини кўрсатади, бунда қолган томонлардан бошқа элементлар ўраб олади. Қиймати — аниқланган тур:
 - **left** — чапга текисланади, ўнгдан бошқалар ўраб олиши мүмкин.
 - **right** — ўнгта текисланади, бошқа элементлар чапдан ўраб олиши мүмкин.
 - **none** — оддий ҳолат (одатий қиймат).
- **clear** — объекти қайси томондан қўшни элементлар ўраб олиш таъқиқланишини кўрсатади. У «float»нинг қийматини бекор қиласди, аниқланган тур қабул қиласди:
 - **left** — чапдаги элементларни пастга туширади.
 - **right** — ўнгда элементлар аксланишини таъқиқлайди.

◦ **both** – икки томондан ҳам ўрашни ман этиб, қаторда ёлғиз ўзи аксланади.

◦ **none** – одатий қиймат, «float» ёки бошқа созлаштарга қараб аксланади.

▪ **outline-color** – ташқи чегара ранги. Одатий қиймати: «invert», ортрангига қараб ўзи аниқлашга имкон беради. Барча браузер буни бирдек кўлламагани учун аниқ ранг бериш тавсия этилади.

▪ **outline-offset** – элементнинг ташқи чегараси (outline) ва ўз чегараси (border) орасидаги масофани ўрнатади. Одатий қиймати: 0. Манфий ўлчамдаги қиймат бериш орқали ташқи чегарани объект ичида аксланишини таъминлаш мумкин.

▪ **outline-style** – ташқи чегара шакли. Қиймати чизик шаклини кидай. [8.9.]

▪ **outline-width** – ташқи чегара қалинлиги. Қиймати «border-width»ники сингари.

▪ **outline** – ташқи чегара шакли. Бирданига «outline-color», «outline-style», «outline-width»ларнинг қийматларини кўрсатишга мўлжалланган. Ташқи чегара билан оддий чегара(border)нинг фарқи шундаки, ташқи чегара объектнинг ўлчамига ва жойлашишига таъсир ўтказмайди ҳамда тўрттала томон учун ҳам бир хилда бўлади.

▪ **z-index** – элементнинг бошқаларга нисбатан устида ёки остида бўлишини англатади. Умуман олганда вебсаҳифадаги барча элементлар битта текислиқда жойлашади, лекин баъзан битта объектнинг бошқаларининг устида уларнинг жойлашувини ўзгартиргмаган ҳолда намоён этишга зарурият туғилади. Бундай ҳолларда керакли объектларни бошқа қатламда акслантиришга тўғри келади. Ушбу хусусият бутун сондаги қиймат қабул қилиб, элементнинг қайси қатламдалигини кўрсатади. Қанча қиймат катта бўлса, демак энг устида жойлашган деган тасаввурни беради. Объектни бошқа қатламда эканлигини «position» хусусияти орқали кўрсатилади.

9.4. Ички элементларини акслантириш

Объектнинг (блокли тегнинг) ичидағи ташкил этувчи элементларини жойлаштириш учун қуийдаги хусусиятлардан фойдаланилади:

▪ **align-content** — оралиқ масофаларни ўзгартыриш орқали «эгилувчан жойлашув»ни таъминлайди. [9.1.]

Қиймати — аниқланган тур:

- **flex-start** — «тахлаш»ни бир бошдан бошлиди;

- **center** — элементларни эгилувчан майдон марказига жойлади;

- **flex-end** — ортиқча бўшилик эгилувчан майдоннинг юқори қисмида бўлади;

- **space-between** — ортиб қолган бўшилик, қаторлар ўртасида тенг тақсимланади;

- **space-around** — барча ички элементларнинг юқори ва пастки ташқи ҳошиялари тенг бўлади;

- **stretch** — ички элементлар ортиқча бўшилик тўлгунча бирдай катталашади;

▪ **justify-content** — элементларнинг ўлчами ва ҳошияларини инобатта олган ҳолда «эгилувчан жойлашув»ни таъминлайди. Қиймати «align-content»ники сингари.

▪ **align-items** — ички элементларнинг тикламасига жойлашувини аниқлайди. Қиймати — аниқланган тур:

- **flex-start** — тепалари бир текисда жойлашади;

- **center** — марказлари ташқи майдон ўртасидаги уфқий чизиқда жойлашади;

- **flex-end** — пастки чегаралари ташқи майдоннинг қуий қисмида жойлашади;

- **stretch** — одатий қиймат, элементлар чўзилиб, тўла баландликни эгаллайди;

- **baseline** — «асос чизиги» бўйича текисланади. [8.8.]

▪ **align-self** — фақат жорий қатордаги элементларни «align-items»да кўрсатилгандагидек жойлаштиради. Одатий қиймати: «auto», яъни ташқи элементи мавжуд бўлмаса, ҳаммасини тенг бўйли қиласи, акс ҳолда «align-items» қийматини олади.

▪ **clip** — элементнинг қандай шаклдаги қисми кўриниши лозимлигини билдирувчи функция кўрсатилади. Ҳозирча фақат тўғри тўргубурчак шакли қўлланилади.

Одатий қиймати «auto» бўлиб, элемент майдони тўлиқ кўринади. Мисол:

```
clip: rect(y1, x2, y2, x1);
```

▪ **column-count** — ички элементларнинг нечта устун сифатида жойлашишини кўрсатади. Бу усул рўзномадагидек матнларни бир неча устун кўринишида ёзиш лозим бўлганда кенг кўлланилади. Одатий қиймати: «auto», ўзи аниқлайди, лекин аниқ сон тариқасида устунлар нечталигини кўрсатиш тавсия этилади.

▪ **column-fill** — обьектнинг ташкил этувчилари устунларга қандай тақсимланишини кўрсатади. (Хозирги браузерлар буни кўлламайди.) Қиймати аниқланган тур:

- **auto** — устунлар кетма-кет тўлдирилиб борилади.

- **balance** — барча устун teng тўлдирилади, фақат чоп этилаётганда ушбу қоида охирги сахифага тадбиқ этилади.

- **balance-all** — худди «balance»дагидек, чоп этилаётганда ҳам барча сахифага тадбиқ этилади.

▪ **column-gap** — устунлар орасидаги масофани аниқлайди. Одатий қиймати: «normal», бунда браузернинг ўзи аниқлайди. Аслида ўлчам сифатида масофа бериш мумкин.

▪ **column-rule** — ушбу умумлаштирилган ҳусусият устунлар орасига чизиқ чизишга мўлжалланган. Одатда чизиқ яширин бўлиб, матн ранги билан бир хил тусда бўлади. Бунда чизиқнинг ранги «column-rule-color», қалинлиги «column-rule-width» ва шакли «column-rule-style» ҳусусиятлари билан [8.9.] алоҳида алоҳида берилиши ҳам мумкин.

▪ **column-span** — ички элемент қанча устунларни эгаллаши лозимлигини кўрсатади. Одатда бу ҳусусият сарлавҳа ёки расм қўйишида кенг кўлланилади. Афсуски хозирча бу ҳусусиятни «Firefox» кўлламайди, устунларга оид юқоридаги ҳусусиятларнинг баъзиларига «-moz-» олд қўшимчаси билан ёзилгандагина уни тўғри акслантиради. Умид қиласизки, келгуси талқинларида буни бошқа браузерлар сингари инобатга олади. Бу ҳусусиятнинг қиймати аниқланган тур:

- **none** — одатий қиймат, битта устунда жойлаши-

шини таъминлайди.

- all – элемент барча устунларни эгаллайди.

▪ **columns** – устунлар сони ва энини бирданига бериш мумкин бўлган умумлаштирилган хусусият. Таъкидлаб ўтиш лозимки, устунларга ажратиш жадвалнинг ичидаги ишламайди, у бошқа блокли теглар учун мўлжалланган. Одатий қиймати: «auto», аслида «**column-width**» ва «**column-count**» қийматларини бўш жой билан ажратилган ҳолда берилади.

▪ **direction** – матн ёзилиш йўналишини кўрсатади ёки элементлар жойлашиш тартибини аниқлайди. Ушбу хусусият ҳар бир ҳарфнинг ўринини тескари тартибда қайтадан жойлаштириб чиқмайди, фақатгина кетма-кетлик қандай йўналишда бўлишини билдиради. Уни жадвалнинг катаклари ёки элементларни аксланиш тартибини ўзгартириш учун ҳам кўллаш мумкин. Қиймати аниқланган тур:

- ltr – чапдан ўнгта, одатий қиймат.

- rtl – ўнгдан чапга.

▪ **object-fit** – қатъий эни ва бўйи кўрсатилган блокли тег ичида расм ёки лавҳа ўлчамини ўзига мослади. Қиймати аниқланган тур:

◦ **fill** – одатий қиймат, объектнинг эни ҳам бўйи ҳам ташқи тегники сингари ўзгартирилиб, тўла майдонни қоплайдиган тарзда акслантирилади.

◦ **contain** – пропорционаллик сақланган ҳолда, эни ёки бўйи ташқи тегникига тенг бўладиган қилиб масштабланади ва ташқи майдон ўртасига жойлашади.

◦ **cover** – пропорционаллик сақланган ҳолда эни ҳам бўйи ҳам ташқи майдонни тўлдирадиган тарзда масштабланади ва сизмай қолган қисми кесиб ташланади.

- **none** – ўлчам ўзгартирилмайди.

▪ **overflow** – ўлчами кўрсатилган блокли элементнинг ичидағилари сифмаган ҳолда қандай йўл тутишни кўрсатади, қиймати аниқланган тур:

◦ **visible** – одатий қиймат, ташқарисига чиқиб кетган қисмлари ҳам кўринади.

- **hidden** – сифматан қисмлари кўринмайди.

- **scroll** – доим айлантиргич кўрсатилади.

- **auto** – ичидағилар сифмаган ҳолларда айлантиргич

күринади.

▪ **overflow-x** — фақат уфқий йўналишда «overflow»ни тадбиқ этади.

▪ **overflow-y** — фақат тик йўналишда «overflow»ни тадбиқ этади.

▪ **vertical-align** — элемент баландлиги ёки ундаги матн қаторларига нисбатан объектнинг текислаб жойлашириди. Қиймати аниқланган тур:

- **baseline** — асос чизигига текислайди. Асос чизиги сифатида матнинг биринчи қаторининг қуи қисми қаралади, агар ундан тасаввурый чизик мавжуд бўлмаса, пастки чегара қабул қилинади (одатий қиймат).

- **bottom** — энг қуи қаторнинг пастки чегарасига текислайди;

- **middle** — асос чизигига элементнинг ўртаси ва унга «х» ҳарфини ярмичалик масофа қўшиб, шу нуқтага нисбатан (тик ярмидан) ўтказилган чизикка текислайди;

- **sub** — асос чизигининг пастида индекс сифатида ўлчами ўзгармасдан ёзилади;

- **super** — асос чизиги тепасида даража сифатида ўлчами сақланиб ёзилади;

- **text-bottom** — сўнги қаторнинг қуи чегарасига текисланади;

- **text-top** — илк қаторнинг юқори чегарасига текисланади;

- **top** — элементнинг юқори чегарасига текисланади;

9.5. Ҳос хусусиятлар

CSSда шундай хусусиятлар мавжудки, улар фақатгина маълум тегларга гина тааллуқли бўлади. Фақат ўзига хос тегнинг хусусиятини ўзgartириш учунгина қўлланилади ва бошқа элементлар учун ишламайди. Масалан:

▪ **caption-side** — жадвал ёки жадвал тарзда акслантирилиши кўрсатилган блокли тегнинг сарлавҳасини қаерда қўринишини аниқлайди. У фақат «table» ичидағи «caption» теги учун қўлланйлади. Қиймати аниқланган тур:

- **top** — одатий қиймат, сарлавҳа жадвалнинг тепа

қисмида кўринади.

- bottom – сарлавҳа жадвалнинг остида кўринади.

▪ **empty-cells** – жадвал катаги ёки жадвал тарзда акслантирилиши кўрсатилган блокли тег чегараси ва орт рангини кўрсатишни таъминлайди. Агар катак ичида ҳеч қандай элемент бўлмаса ёки аксланмайдиган бўш жойлар мавжуд бўлса, у бўш катак ҳисобланади. Ушбу хусусиятнинг қиймати «**border-collapse: collapse;**» берилганда эътиборсиз қолдирилади. [8.9.] Қиймати:

- show – одатий қиймати, ортранг ва чегаралар кўринади. Бу HTML5 талаби бўйича, олдин аксинча эди ва бўш катак кўриниши учун аксланувчи бўш жой – « » қўйиб ёзишга тўғри келарди.

◦ hide – катак кўринмайди, ҳудди «**visibility: hidden;**» берилгани каби. [8.9.]

▪ **table-layout** – жадвал катаги ёки жадвал тарзда акслантирилиши кўрсатилган блокли тег объектининг ўлчамини қандай аниқланишини белгилайди. Қиймати аниқланган тур:

◦ auto – Браузер жадвалнинг тўлалигича юклайди, таркибини таҳлил қилиб катаклар ўлчамлари қандай бўлишини аниқлайди ва ундан сўнгтина акслантиради.

◦ fixed – Катаклар эни биринчи қатордагилар учун кўрсатилган ўлчамда ёки **<col>** тегида кўрсатилгани каби бўлади. Агар кенглик кўрсатилмаган бўлса, катакларнинг эни тенг бўлади. Ёзувлар катак ичига тўғри жойлашиши учун бундай жадвалнинг ўлчамини олдиндан беришга тўғри келади.

▪ **quotes** – қўштироқ қўйиши таъминлайди. У фақат **<q>** теги ичида ёзилган матнни қўштироққа олади. Маълумки, қўштироқнинг очилиш ва ёпилиш белгилари бир-биридан фарқ қиласди, улар бўш жой билан ажратилиб кўрсатилади. Махсус код ёки белгиланиши ёрдамида ҳам ёзиш мумкин. Мисол:

```
quotes: &laquo; &raquo;
```

▪ **resize** – фойдаланувчига **<textarea>** элементининг ўлчамини ўзгартириш имкониятини яратади. Қиймати

аниқланган тур:

- **none** – ўлчам ўзгартириш мүмкін әмас;
- **both** – ҳар иккі йұналишда ўлчамини ўзгартириш мүмкін;
- **horizontal** – фақат уфқий йұналишда ўлчамни ўзгартириш мүмкін;
- **vertical** – фақат тик йұналишда ўлчамни ўзгартириш мүмкін;
- **tab-size** – табулятор нечта белги әнича ўлчамни әгаллашини күрсатади. Қиймати бутун сон, одатда 8 қабул қиласы. Лекин оддий матнда у фақат биттә бүш жой билан күрсатылады. Ушбу хусусият қандай тадбиқ этилишини күриш учун `<pre>` элементи ичидаги матнга қарааш лозим.

9.6. Ички маттінни жилолаш

Объект ичидаги маттіннинг қандай күринищда намоён бўлишини қўйидаги хусусиятлар ёрдамида бошқариш мүмкін:

- **color** – Матн ранги, қиймати: ранг.
- **hyphens** – белгиланган кенгликка сифмаган матндағи сўзларни кейинги қаторга қандай тарзда бўғинлаб туширилишини аниқлайди. Қиймати аниқланган тур:
 - **none** – маҳсус «`­`» белгилари қўйилган бўлса ҳам бўғин кўчирилмайди;
 - **manual** – одатий қиймат, фақат «`­`» белгиси ёки «`wbr`» теги қўйилган жойларда бўғин кўчирилади.
 - **auto** – браузернинг ўзи аниқлайди. Бунда «юмшоқ кўчириш»лар дан ташқари күрсатылган тил учун браузернинг ўзида мавжуд бўғин кўчириш шартлари ҳам инобатга олинади.
- **letter-spacing** – матндағи ҳарфлар орасидаги масофани ўзгартыради. Одатий қиймати: «`normal`», ўлчам сифатида оралиқ масофани күрсатыш мүмкін. Ҳарфнинг катталыгигита мос равища «`em`» ёки «`ex`» да қиймат беріш тавсия этилади.
- **line-height** – сатрнинг энг кичик баландлигини аниқлайди. Агар сатрда `` теги қатнашган бўлса,

ушбу хусусият аҳамиятини йўқотади. Қийматини ўлчам ёки фоизда кўрсатиш мумкин. Одатий қиймати **normal** бўлиб, браузернинг ўзи аниқлагандай намоён бўлишини таъминлади.

▪ **font-family** — хусниҳат номи. Бир нечта ном вергул билан ажратилиб кўрсатилади. Агар ном бир нечта сўздан иборат бўлиб, бўш жой билан ажратилса, уни қўштириноқ ичида ёзилиш керак. Рўйхатдаги номлардан биринчи мавжуд бўлган хусниҳат тадбиқ этилади. Одатий қиймати браузерда белгиланади, кўп ҳолларда у «Times New Roman» бўлади.

▪ **font-kerning** — ҳарфлар орасидаги маҳсус масофа. Агар ҳамма ҳарф орасида бир хил масофа бўлса, икки томонга эгалиб ёзиладиган ҳарфлар орасидаги масофа бошқаларига нисбатан каттадай туюлади. Масалан: «УЛ» ёзилганда «У»нинг тепаси ва «Л»нинг пастки қисми битта тик чизикда жойлашган, «ИШ» сўзида эса бундай масофа кўйиб бўлмайди. Ушбу маҳсус оралиқ қандай кўйилиши хусниҳатнинг ўзида аниқланган бўлади. Қиймати аниқланган тур:

○ **auto** — агар хусниҳатда маҳсус оралиқни кўйиш имконияти мавжуд бўлса, бу ҳолда ўлчам кичик бўлган ҳолларда ўқиш қулий бўлиш учун уни учиради. Бошқа ҳолларда ўзи созлаб меъёридаги оралиқни ўрнатади.

○ **normal** — доим маҳсус оралиқ тадбиқ этилади.

○ **none** — маҳсус оралиқ қўлланилмайди.

▪ **font-size** — хусниҳат ўлчами. Ўлчамни маҳсус сўзлар билан ҳам ифодалаш мумкин: xx-small, x-small, small, medium, large, x-large, xx-large. Булар HTMLдаги 1 дан 7 гача кўрсатиладиган қиёсий қийматларнинг ўрнига қўлланилади.

▪ **font-stretch** — ҳарфларнинг эни қандай жой эгаллашини аниқлайди. Қиймати аниқланган тур бўлиб, қийидаги қиёсий сўзлар билан берилади: ultra-condensed, extra-condensed, condensed, semi-condensed, normal, semi-expanded, expanded, extra-expanded, ultra-expanded.

▪ **font-style** — хусниҳат шакли. Қиймати аниқланган тур:

○ **normal** — одатий кўринишда;

○ **italic** — ёзма ҳарфлар билан ифодаланган;

- **oblique** – ҳарфлар ўнга оғдирилган шаклда акслантирилган;
- **font-variant** – ёзувни бош ҳарфларда ифодалашни таъминлайди, қиймати:
 - **normal** – одатдагидек кўринишни сақлади;
 - **small-caps** – барча кичик ҳарфларни бош ҳарфлар ёрдамида ифодалайди, аммо ўлчами ўз ҳолиша (кичиклигича) қолади;
- **font-weight** – ҳарфларни ифодаловчи чизиқларнинг қалинлиги ва рангининг тўқлик даражасини аниқлайди. Қиймати «bold», «bolder», «lighter», «normal» каби қиёсий сўзлар ёрдамида ёки 100дан 900гача юзлик сонлар билан берилиши мумкин.
- **font** – кенгайтирилган хусусияти юқорида келтирилган хуснихатга доир хусусиятларни бирданига кўрсатишга имкон беради. Ўлчамини кўрсатилаётганда сатрнинг баландлигини ҳам бериш мумкин. Умумий синтаксиси қўйидагича:

```
font: [font-style font-variant font-weight font-stretch]  

        font-size [/line-height] font-family
```

- **list-style-image** – рўйхатга нишонча сифатида расм бириктиради. Қийматига файл манзилини **url** функцияси орқали берилади.
- **list-style-position** – рўйхат нишончасини жойлашиш ўрнини белгилайди. Қиймати аниқланган тур:
 - **inside** – нишонча матн бошланадиган нуқтага қўйилади, яъни у иккинчи қатордан бошлаб кейинги сатрлар билан чапдан бир текис жойлашади.
 - **outside** – одатий қиймат, матннинг сатрлари текисланган чап томондан нишонча олдинга ажралиб чиқиб турган ҳолда жойлашади.
- **list-style-type** – нишончанинг кўриниши. Қиймати аниқланган тур:
 - **circle** – айлана сифатида;
 - **disc** – юмалоқ нуқта сифатида (**** теги учун одатий қиймат);
 - **square** – квадрат шаклида.
 - **armenian** – анъанавий арман рақамлари бўйича;

- **decimal** – араб рақамлари билан (1, 2, 3, 4, ...);
 - **decimal-leading-zero** – араб рақамлари ёрдамида, фақат ўндан кичик сонларнинг олди нол билан тўлдирилади (01, 02, 03,...);
 - **georgian** – анъанавий грузин рақамлари билан;
 - **lower-alpha** – кичик лотин ҳарфлари орқали (a, b, c, d,...);
 - **lower-greek** – кичик грек ҳарфлари ёрдамида (α, β, γ, δ,...);
 - **lower-latin** – «lower-alpha» сингари;
 - **lower-roman** – кичик рим рақамлари орқали (i, ii, iii, iv, v,...);
 - **upper-alpha** – бош лотин ҳарфлари билан (A, B, C, D,...);
 - **upper-latin** – «upper-alpha» сингари;
 - **upper-roman** – бош рим ҳарфлари орқали (I, II, III, IV, V, ...);
 - **none** – нишонча қўйилмайди.
- **list-style** – рўйхат шаклини аниқлайди, бу кенгайтирилган хусусиятда «list-style-type», «list-style-position» ва «list-style-image»ларнинг қийматларини умумлаштириб кўрсатиш мумкин. Рўйхат сифатида `<dd>`, `<dt>`, ``, `` ва `` теглари ичидағи матн ёки «display: list-item» қийматини олган обьект қаралади.
- **opacity** – шаффоффлик, одатий қиймати: 1, унда ҳеч ўзгаришсиз аксланади. Қиймат сифатида шаффоффли дараҷасини кўрсатадиган 0 дан 1 гача ҳақиқий сон берилади. 0 буткул шаффоффликнинг англатиб, кўринмайдиган даражада тиник ойна мисоли бўлади. Хираки шаффоффлик билан чалкаштирмаслик керак. Хира элемент орқасида нима борлиги кўринмайди, шаффофф элемент қанчалик тиник бўлса, орқасидаги обьектлар шунчалик яққол кўринади.
- **text-align** – матннинг уфқий текисланишини аниқлайди. Қиймати аниқланган тур:
- **center** – матн қаторнинг ўртасига нисбатан симметрик тарзда жойлашади;
 - **justify** – сўзлар орасидаги бўш масофа кенгайтирилиб, матннинг ўнг ва чап томони бўйича текис жойлаштирилади;

- **left** – фақат чап тараф бүйича текис жойлаштирилади (одатий қиймат);
- **right** – фақат ўнг тараф бүйича текис жойлаштирилади;
- **start** – матн йўналиши чапдан ўнгта бўлса чапга; аксинча – ўнгта текислайди;
- **end** – «start»нинг акси («dir»нинг қийматига боғлик).
- **text-align-last** – агар «text-align: justify» қиймати берилган бўлса, охирги қаторни қандай текислаш кераклигини кўрсатади. Қиймати аниқланган тур:
 - **start** – сатрнинг бошлангич томонига нисбатан текислайди (одатий қиймат);
 - **end** – якуний томонига нисбатан текислайди («dir»нинг қийматига боғлик);
 - **left** – айнан чап томонга қараб текислайди;
 - **right** – айнан ўнг томонга нисбатан текислайди;
 - **center** – сатрни ўртага текислайди;
 - **justify** – кенглик бўйича ҳар икки томонга нисбатан текислайди, агар фақат битта сўздан иборат бўлса у чап томонда бўлади;
- **text-decoration-line** – матннинг қаерида чизик чизилишини аниқлайди. Қиймати:
 - **line-through** – матн (ҳар бир ҳарф) устидан чизилади;
 - **overline** – тепасидан чизилади;
 - **underline** – матн остидан чизилади;
 - **none** – одатий қиймат, чизикни бекор қилади (ишорат учун ҳам тааллуқли);
- **text-decoration-style** – матн чизигининг («text-decoration-line» орқали берилган) шакли. Қиймати аниқланган тур:
 - **solid** – битта тўғри чизик;
 - **double** – иккита параллел тўғри чизик;
 - **dotted** – нуқталардан ташкил топган чизик;
 - **dashed** – узук чизик;
 - **wavy** – тўлқинли чизик;
- **text-decoration-color** – «text-decoration-line» орқали берилган чизик ранги. Одатий қиймати матн ранги билан бир хил.

▪ **text-decoration** — кенгайтирилган хусусият «text-decoration-line», «text-decoration-style» ва «text-decoration-color»лар қийматларини бирданига бериш имкониятини яратади.

▪ **text-indent** — хатбошини қолган қаторларга нисбатан қанча ичкарида бўлишини кўрсатади. Қиймати ўлчам ёки фоиз хисобида берилади.

▪ **text-overflow** — блокли элемент ичидағи матн кўрсатилган ўлчамга сифмаган тақдирда уни қандай кесиб кўрсатишни ифодалайди. Ушбу хусусият ўз-ўзидан «overflow: visible» берилган ҳолда ишламайди. Йкки хил қиймат қабул қиласди:

○ **clip** — матннинг майдонга сифадиган қисми кесиб кўрсатилади ҳолос;

○ **ellipsis** — матн майдонга сифмаса, кесиб охирига уч нуқта қўшилади;

▪ **text-shadow** — матннинг сояси. Одатий қиймати: «none», у соясиз кўринишни таъминлайди. Умуман олганда соя қиймати қўйидагича берилади:

Синтаксис:	<code>text-shadow: x y r c, x y r c</code>
Мисол:	<code>text-shadow: 1px 1px 2px black, 0 0 1em red;</code>

Бунда x — x ўқи бўйича силжиш; y — y ўқи бўйича силжиш; r — соя ёйилиш радиуси; c — ранги; Бир неча қийматлар вергул билан ажратиб берилади.

▪ **text-transform** — матнни бош ёки кичик ҳарфли ёзувга айлантиради. Қийматлари:

○ **capitalize** — ҳар бир сўздаги биринчи белгини бош ҳарфда ёзилишини таъминлайди, қолган ҳарфлар ўз ҳолида қолади;

○ **lowercase** — матн фақат кичик ҳарфлар билан ёзилади;

○ **uppercase** — матн фақат бош ҳарфлар билан ёзилади;

○ **none** — одатий қиймат, ўз ҳолида намоён бўлади;

▪ **unicode-bidi** — «direction»да кўрсатилган йўналиш бўйича ҳар бир ҳарфларнинг ўрнини ўзгартириб акслантиради. Масалан, араб тилида ўнгдан чапга қараб ёзилади. Бунинг учун матнни қайтадан териб чиқиши

шарт эмас, айниқса дастурий таъминот билан таржима қилиниб ёзилганда. Қиймати аниқланган тур:

- **normal** — одатий қиймат, браузернинг ўзи юни-код белгилари бўйича аниқлайди.
- **embed** — худди «direction» кўрсатилгандай аксланинади. (Хатбоши охиридаги нуқта қатор бошига чиқиб қолади)
- **bidi-overide** — «direction»да кўрсатилган йўналиш бўйича ҳар бир ҳарф ўрни алмаштирилади.
- **white-space** — бўш жойларни қандай аксланишини кўрсатади. Қиймати:

◦ **normal** — одатий қиймат, барча бўш жойлар ва қаторларни ажратувчилари ягона бўш жой сифатида ифодаланади. Қаторларга бўлишни ўзи аниқлайди;

◦ **nowrap** — қаторларга бўлиш бўш жойлар бўйича амалга оширилмайди фақат **
** теги инобатга олиниади;

◦ **pre** — қандай ёзилган бўлса, шундайлигича на-моён бўлади;

◦ **pre-line** — эни сифмаса қоторларга бўлинади ва ҳар бир қаторнинг боши ва охиридаги бўш жойлар учирилади;

◦ **pre-wrap** — ёзилгани каби аксланади, фақат эни бўйича кейинги қаторга туширилиши мумкин;

▪ **word-break** — сўз бўлагини кейинги қаторга туширишга рұхсат бериш. Қиймати:

◦ **normal** — одатий қиймат, сўз бўлагизи кейинги қаторга тушишига рұхсат бермайди, агар аниқ **
** теги ёки ажратувчи белги қўйилмаган бўлса;

◦ **break-all** — матн кўрсатилган кенгликка сифиши учун сўзни бўлакларга ажратишни ўзи аниқлайди;

◦ **keep-all** — кейинги қаторга сўзнинг бўлаги тушишини хитой, корейс ва япон тиллари учун таъқиқлайди. Колган ҳолларда «normal» сингари ишлайди.

▪ **word-spacing** — сўзлар орасидаги масофани аниқлайди. Қиймати: ўлчам. Мабодо «text-align: justify» берилиган бўлса ҳам, сўзлар орасидаги масофа ушбу хусусиятда кўрсатилгандан кичик бўлмайди.

▪ **word-wrap** — маъно жиҳатдан «word-break» билан бир хил вазифа бажаради. Фарқи шундаки, «word-

`break`»да тўлалигича жорий қаторга сиғмаган сўзниңг ўзи кейинги қаторга олинмай, унинг қисмигина кўчирилади; «`word-wrap`»да эса аввал сўзниңг ўзи туширилади, шунда ҳам кенглик етмаган ҳолда унинг қисми кейинги қаторга ёзилади. Қиймати аниқланган тур:

- `normal` – фақат `
` теги ёки унга ўхшаган ажратувчилар қўйилганда гина кейинги қаторга ўша жойдан кўчирилади;
- `break-word` – берилган кенгликка қараб сўз ёки унинг қисми туширилади;
- `writing-mode` – ёзувнинг тик ёки уфқий бўлишини таъминлайди. Қиймати:

- `horizontal-tb` – одатий қиймат, уфқий чапдан ўнга ва тепадан пастга йўналган;
- `vertical-rl` – тик тепадан пастга ва ўнгдан чапга;
- `vertical-lr` – тик тепадан пастга ва чапдан ўнга;
- `sideways-rl` – тик тепадан пастга, ҳамма ҳарфлар ўнгта «ёнбошлайди»;
- `sideways-lr` – тик тепадан пастга, ҳамма ҳарфлар чапга «ёнбошлайди»;

9.7. Чоп этиш хусусиятлари

Чоп этиш қурилмалари турлича бўлганилиги боис, кўпинча, вароқнинг четларидан турлича жой ташлашга ёки ўзгача ўлчамда акслантиришга тўғри келади. Чоп этилаётган саҳифа учун қонуниятлар «`@page`» калит сўзи ёрдамида берилади. Ундан кейин қонуниятлар қайси саҳифалар учун тадбик этилиши кўрсатилиши мумкин: «`:first`» – фақат биринчи саҳифага; «`:left`» – чап (жуфт ўриндаги) саҳифаларга; «`:right`» – ўнг (тоқ ўриндаги) саҳифаларга. Мисол:

```
@page :first { top: 3.5sm; }
```

Баъзи хусусиятлар фақат чоп этилаётганда гина иnobatga олинади. Масалан:

- `marks` – вароқларга бўлувчи белги ўрнатади. (Таъкидлаб ўтиш лозимки, ҳозирча бу ажойиб хусусиятни браузерлар кўлламайди.) Қиймати аниқланган тур:
 - `crop` – обьект чоп этилгандан сўнг навбатдагиси-

ни кейинги вароқقا туширади;

◦ **cross** — кейинги вароқдаги обьект ҳам жорий-сини бир бўлагилигини англатади. Мисол учун, бир нечта вароқقا битта катта расмни чоп этиш.

◦ **none** — одатий қиймат, ажратувчи қўйилмайди.

▪ **orphans** — бир нечта вароқقا жойлашадиган матнлар учун олдинги вароқда камида қанча қатор чоп этилиши кераклигини кўрсатади. Қиймати: бутун сон.

▪ **page-break-after** — чоп этиш жараёнида кўрсатилган элементдан сўнг кейинги обьектларни бошқа вароқقا чиқарилишини кўрсатади. Қиймати аниқланган тур:

◦ **always** — доим вароқларга ажратилишини таъминлайди;

◦ **auto** — ажратиш фақат зарур бўлгандагина бажарилади (одатий қиймат);

◦ **avoid** — элементдан сўнг ажратиласдан чоп этилишини кўрсатади;

◦ **left** — кейингиси жуфт ўринда бўлиши учун бир ёки икки вароқ бўш чиқаради;

◦ **right** — кейингиси тоқ ўринда бўлиши учун бир ёки икки вароқ бўш чиқаради;

Ҳар бир уфқий чизиқдан кейин янги вароқдан чоп этишни таъминловчи мисол:

```
hr { page-break-after: always; }
```

▪ **page-break-before** — кўрсатилган элементдан бошлаб кейинги вароқда чоп этилишини таъминлайди, қиймати «page-break-after»ники сингари.

▪ **page-break-inside** — кўрсатилган элемент турли вароқларда чоп этилиш ҳолатини аниқлади. Қиймати «**avoid**» бўлса, элемент фақат битта вароқда чиқишини таъминлайди. Одатий қиймат «**auto**» берилса, зарур бўлса бўлинниб чоп этилади.

▪ **widows** — бир неча вароқда чоп этиладиган саҳифа учун навбатдаги вароқда энг камида нечта қатор бўлиши лозимлигини кўрсатади. Ушбу хусусият қиймати бутун сон бўлиб, устивор ҳисобланади ва у «**orphans**»никига зид бўлса ҳам инобатта олинади.

9.8. Ҳодисаларни бошқариш

- **pointer-events** — сичқонча босилгандағи ҳодисаны бажарилишини бошқаради. Қиймати аниқланған түр:
 - **auto** — ҳодиса одатдагидек бажарилади;
 - **none** — ҳодиса бажарилишини тұхтатади;
- **user-select** — фойдаланувчи томонидан матнни белгилашни бошқаради. Баъзида сарлавҳани англатувчи ёки алоҳида аҳамият касб этувчи матнларни (нусха олиш учун) белгилаш мақсадға мувофиқ әмас, шундай холларда ушбу хусусиятни күллаш мумкин. Аммо күп браузерлар ҳозирча буни тұғридан-тұғри құлламайды, «Internet Explorer»га «-ms-user-select»; «Mozilla Firefox»га «-moz-user-select»; қолғанлари учун эса «-webkit-user-select» каби ёзиш тақазо қилинади. Қийматлари:
 - **auto** — ташқи элемент хусусиятига қараб олинади;
 - **none** — белгилаш таъқиқланади;
 - **text** — факт матнни белгилаш мумкин бўлади;
 - **all** — бутун обьектни ички элементлари билан тўлалигича белгилайди;
 - **contain** — факт элемент чегарасининг ичидаги белгилашта рухсат беради;

9.9. Вазиятан синфлаш

CSSда вазиятан синфлаш бир қараңда табақалашта үшшаб кетса-да, моҳияттан үндан кескин фарқ қиласы. Табақалаштиришда айни пайтдаги ҳолатига қараб шамойил берилади, вазиятан синфлаш эса факт майлум вазиятлардагина ёки элементнинг ажратилиган майлум бўлакларигагина таъсир қилувчи шакл-сиймо берилади. Уларни фарқлаш учун табақалаштиришда «::» ишлатилса, вазиятан синфлашда «::» қўлланилади. Масалан, **content** хусусияти фактгина элементнинг иккиси томондан чегарасини англатувчи «::after» ёки «::before» вазият синфлари билангина қўлланилади. Бу йккиси мөсравишида обьектнинг ортидан ёки олдидан қўшімчә жило беришда фойдаланилади. Факт шу иккаласи учун кет-

ма-кет иккита икки нуқта ўрнига биттасини қўйса ҳам бўлади.

Агар қандайдир рўйхатга махсус жозиба бериш билан бирга унга тартиб рақами ҳам кўшиш лозим бўлса, уни CSS имкониятлари ёрдамида ҳам бажариш мумкин. Бунинг учун рўйхатни ифодаловчи тегини қўллаш ҳам талаб этилмайди. Ушбу имкониятни амалга ошириш учун хизмат қиладиган хусусиятлар билан қўйида қисқача танишиб ўтамиз:

▪ **counter-reset** — саноқ сақланадиган ўзгарувчини аниқлайди. Қиймати:

◦ none — саноқ амалга оширилишини таъқиқлайди;

◦ inherit — саноқ қиймати ташқи элементдагидан олинади;

◦ идентификатор сон — саноқ сақланадиган ўзгарувчининг номи ва бошланғич қиймати. Бир нечта кўрсатилиши лозим бўлса, бўш жой билан ажратилади.

▪ **counter-increment** — саноқнинг кейинги қийматини ҳисоблайди. Қиймати:

◦ none — саноқни оширилишини таъқиқлайди;

◦ идентификатор сон — кўрсатилган ўзгарувчи қийматини берилган сонга оширади.

▪ **content** — тег ичидағи матнга қўшимча равища акслантириладиган ёзув. Қиймат сифатида ихтиёрий сатр берилиши мумкин. Агар тегнинг бирор аломатининг қийматини кўрсатиш лозим бўлса, attr (alomat) функциясидан фойдаланилади. Кўштирнок қўшиш лозим бўлса уни айнан ёзиш ўрнига «open-quote» ёки «close-quote» сўзларидан фойдаланиш мумкин. Сатр қиймат қайтарадиган функциялар ишлатилса, уни кўштирнокка олинган оддий сатр билан ҳеч қандай математик амалсиз бириктириш мумкин.

Махсус функция counter ёрдамида кўрсатилган ўзгарувчининг жорий қиймати олинади. Энди шу маълумотлар ёрдамида саноқли рўйхат тузамиз:

HTML	<pre><div> <h1>HTML 5</h1> <h1>CSS 3</h1> <h1>Веб-саҳифа</h1> </div></pre>
CSS	<pre>div { counter-reset: n 1; } h1::before { counter-increment: n 1; content: counter (n) "-бўлим.";</pre>
Натижа	2-бўлим. HTML 5 3-бўлим. CSS 3 4-бўлим. Веб-саҳифа

Ушбу мисолда ҳар бир `<h1>` тегидаги ёзув олдидан нечанчи бўлимлигини қўшиб чиқдик. CSS – HTMLда энг кам код ёзишни таъминлаб берувчи воситадир. Шундай экан, бизга қулайлик яратувчи яна бир нечта вазият синфлари билан танишиб чиқамиз:

◆ **::first-letter** – Матннинг фақат биринчи ҳарфига тадбиқ этиладиган синф. Айрим рўзномаларда биринчи ҳарфлар алоҳида ажратиб кўрсатилишига гувоҳ бўлганмиз. Шундай талаб қўйиладиган бўлса, фақаттина CSS ёрдамида буни амалга ошириш мумкин. HTMLни таҳрирлаш ҳар доим ҳам осон кечавермайди, айниқса у бошқа дастурий таъминотлар ёрдамида ҳосил қилинган бўлса.

◆ **::first-line** – биринчи қаторга тадбиқ этилади. Бу қонун ҳужжатларида модда рақами ва номини ажратиб кўрсатишда қўл келади. Умуман олганда кенг тадбиқ этиш мумкин.

◆ **::selection** – матннинг белгиланган қисмигагина тадбиқ этилади. Таъкидлаб ўтиш лозимки, ушбу синф CSS3дан чиқариб ташланган бўлиб, уни ривожлантириш CSS4 учун қолдирилган. Ҳозирча «Mozilla Firefox» буни тўғридан-тўғри кўлламайди, унинг учун алоҳида «`::-moz-selection`» сифатида ёзиш керак. Ўмид қиласиз, бу нокулайлик яқин вақтларда ҳал этилади. Агар (нусҳа олиш учун) белгиланган матннинг

бўлаги бошқача тарзда кўриниб туриши лозим бўлса, қуидагича қонуният ёзишга тўғри келади:

```
p::selection {  
    background-color: #62BA21;  
    color: white;  
}  
p::-moz-selection {  
    background-color: #62BA21;  
    color: white;  
}
```

«Internet Explorer» ўзи учун бир қанча вазият синфларини киритган, аммо бошқа браузерлар уни қўлламагани ва ҳатто Microsoft ҳам у боши берк кўчага кириб қолгани учун бошқа «Edge» браузери ишлаб чиққани боис, биз унинг синфларига тўхталиб ўтишни лозим топмадик.

Х БҮЛІМ

МУКАММАЛ САЙТ ЯРАТИШ ТАЛАБЛАРИ



Мукаммал сайт яратиши талаб этилар экан, ёзилған код барча браузерларда бирдей намоён бўлиши шарт. HTML4 да бу жуда катта муаммо бўлгани боис, маҳсус «ДОСТУРЕ» киритилган эди. Аммо у ҳам самарали чора эмаслиги амалда кўринди. Ўн йиллик музокаралардан кейин ҳаётий тажрибалар асосида яратилган HTML5 чиққанига ва оммавий қўлланилаётганига қарамасдан, ҳалигача баъзи қонуниятлар айрим браузерларда кутилганидек аксланмаяпти. Шундай экан, демак уларнинг ўрнига универсал усульнини топиб қўллашга тўғри келади. Ушбу талабни инобатга олган ҳолда, юқоридаги танишиб ўтган билимларимиз асосида, веб-сайтлар яратишда кенг қўлланиладиган бир неча тажрибалар билан кўйида танишиб ўтамиз.

10.1. Бехато код ёзиш

Юқорида таъкидлаганимиздай, HTML ва CSS дастурлаш тили бўлмаганилиги боис, хато ёки нуқсонли ёзилған кодларни браузер имкон қадар таҳлил этиб муносиб қонуният бўйича намойиш этади. Умуман иложи бўлмаган ҳолда эса, эътиборсиз қолдиради. Лекин бу маъсуллиятсизлик билан код ёзишга ёндашишга рухсат бермайди. Мукаммал дастурлашни ўрганишга бел боғлаган эканмиз, коддаги ҳар бир белги нима учун қўйилганини англашимиз ва унинг ўринли эканлигига ишончимиз комил бўлиши керак.

Агар олдиндан яратилган сахифани такомиллаштириш лозим бўлса, аввал уни барча қоидалар билан тўғри ёзил-

гандигини текшириб кўриш, сўнгра эса қўшимчалар киритиш тавсия этилади. Акс ҳолда топиш жуда мушкул бўйган хатоликлар, чалкашликлар келиб чиқиши мумкин. Масалан, битта файлда қайсиdir блокли элементнинг ёпувчи теги кўрсатилмаган, унинг ўрнига янгишиб яна тегнинг очиш қисми ёзилган. У файлни акслантирилаёттандা фойдаланувчи бу камчиликни сезмайди, лекин бу код тўғри якунланган деб ҳисоблаган ҳолда давомидан бошқа файлдаги кодларни унга бирлаштирилганда бузилиш юзага келади. Буни оддий математик аммалларни бажаришда қавсларни тўғри қўймаслик оқибатида ҳисоб натижаси турлича чиқиши билан қиёслаш мумкин.

Код тўғри ёзилганинги текшириш учун браузерларнинг ўз имкониятларидан ёки уларга қўшимча юкланидиган «Html Validator» кенгайтмасидан фойдаланиш мумкин. Интерфейси содда ва турли браузерларда турлича бўлганлиги боис, қандай фойдаланишини батафсил ёритишга тўхталиб ўтмаймиз. Агар вебсаҳифа интернетда жойлашган бўлса, [«http://validator.w3.org/»](http://validator.w3.org/) сайти орқали ҳам текшириш мумкин.

Код ёзувчиларнинг беэътиборлиги ёки лоқайдлиги ёхуд мукаммал билимга эга эмаслиги боис энг кўп йўл қўядиган камчиликларини санаб ўтишни лозим топдик:

- ◆ **DOCTYPE кўрсатилмаган.** HTML файлнинг биринчи қатори «DOCTYPE» кўрсатишдан бошланиши керак. [2.2.] [2.3.] Акс ҳолда HTML5 қонуниятлари тўғри қўлланымаслиги, баъзида HTML4 тадбиқ этилиши мумкин.

- ◆ **Имло хато.** Тегнинг номи ёки аломати нотўғри ёзилган. Одатда бундай ёзувлар эътиборсиз қолдирилади, лекин AngularJS фойдаланувчилари биладики, объектга мурожаат қилиш ёки унинг кўринишини ўзгартириш учун унга маҳсус номдаги HTMLда мавжуд бўлмаган аломатлар атайин ёзилади. Албатта уни ишлатувчилар етарлича, аммо ўз қоидаларини сингдириш учун бундай «бузгунчилик» қилиш мақсадга мувофиқ эмас.

- ◆ **Мажбурий аломат.** Баъзи тегларнинг маълум аломатлари доим кўрсатилиши шарт. Масалан, «link» теги ишорат кўрсатиш учунгина қўлланилади, унда «href» бўлиши керак.

◆ **Нотаниш ном**. Баъзи теглар ёки аломатлар асли-да HTML қонуниятларида мавжуд әмас, лекин талабга қараб браузерларнинг ўзлари киритишган. Кейинчалик оммаласиб кўп бошқа браузерлар ҳам қўллай бошли-ган. HTML5да уларнинг бир қанчаси янгилик сифати-да қўшилган бўлсада, айримлари CSS орқали ифода-лаш мумкин бўлганлиги боис «тан олинмаган». Масалан Microsoft киритган «tagquee» [8.7.] теги югурувчи сатрни ифодалаш учун барча браузерларда ишлайди, аммо но-таниш ном сифатида қўллаш тавсия этилмайди.

◆ **Ноўрин тег**. Юқорида такидлаганимиздай, қаторли тег ичига блокли тег ёзиш тавсия этилмайди. Бундан ташқари бир неча теглар кетма-кет ёзилаётганда ёпувчи қисмлари очувчисига мутаносиб бўлиши керак. Яъни, олдин очилган – кейин ёпилади. Ноўрин ёзиладиган тегларнинг энг даҳшатлиси битта объектни яратиш учун қатнашадиган теглар ўз ташки теги ичидаги келмас-лигидир. Масалан жадвалда катақни ифодаловчи «td» қаторни ифодаловчи «tr» теги ичидаги бўлиши шарт.

◆ **Ёшилмаган тег**. Кўп тегларнинг ёпувчи қисми кўр-сатилиши шарт бўлмаса ҳам уни ёзиш тавсия этилади. Акс ҳолда қўшни шу номдаги тег билан аралашиб ке-тиши мумкин.

◆ **Киймат бериш**. Кийматлар сон ёки бўш жой ва ажратувчи белги ишлатилмаган сатрдан иборат бўлса, ало-матта тенглаш пайтида қўштирноқ ишлатмаса ҳам бўли-шини айтиб ўтган эдик. [2.1.] Лекин барча кийматларни қўштирноқ ичидаги ёзиш қатъий талаб этилади. Сабаби файлнинг кодламаси ўзгарганда баъзи юникод белгилари ажратувчи сифатида қабул қилиниши мумкин. Ун-дан ташқари бир жойда қўштирноқ қўйиб, бошқа жойда қўймаслик нафақат чалкашлик келтириб чиқаради, балки коднинг чиройли ёзилишига ҳам путур етказади.

◆ **Махсус белги**. Махсус белгилар «&» белгиси ёрда-мида рақамли коди ёки қисқа номи орқали ифодала-нади. [2.1.] Охирида эса «;» ёзилмаса ҳам кутилгандай ўтириб олиш амалга ошсада, аслида уни қўйиш талаб этилади.

◆ **Такрорланниш**. Кўпинча техник хатолик туфайли бит-

та номни қайтта-қайта ёзилб юборилади. Масалан: «****» каби ёзувлар ортиқча тақрорланишдан бошқа нарса эмас. Зеро, қалин ёзувни таъминлаш учун бу каби теглар кетмакет ёзилмайди. Ёки тегнинг ичида айни номдаги алматига бир неча бор қиймат бериш ҳам ўринсиздир! Одатда бундай ҳолларда охирги кўрсатилган қиймат инобатта олиниади, лекин бу ҳам аниқ қоида эмас.

♦ **Ортиқча тег**. HTMLда бир-бирига ўхшаш саҳифалар ярататтеганимизда албатта, нусҳа кўчиришдан фойдаланамиз, баъзан бир неча жойдан нусҳаланган кодлар ичида умумий олиб қараганда ортиқча ҳисобланадиган теглар келиб қолиши табиий. Ёзилган кодни албатта назоратдан ўтказиш керак. Масалан «body» ва «head» каби теглар битта веб саҳифа учун ягона «html» теги ичида фақат биттадан қатнашиши шарт. Ёки ягона «div» кутилгандағидек натижа бера оладиган вазиятда уни яна бир неча блокли тегга олиш самараасиз ва фақат браузер ишини қийинлаштиради холос.

♦ **Ноўрин ёзув**. Кейинги китобларимизда HTML файлни бошқа дастурий таъминотлар орқали яратиш, унинг кодларини маҳсус буйруқлар ёрдамида ҳосил қилишни кўриб ўтамиз. Бунда арзимаган янглишиш туфайли, теглар ёзилиши ва жойлашиши ўзгариб кетаверади. Аслида барча кўринадиган объектлар «body» тегининг ичида жойлашиши шарт [2.5.], унинг ташқарисидаги ёзувлар ҳам айрим браузерларда акслантирилганлиги боис, баъзилар эътибор қаратмаслиги мумкин. Дастур тузиш жараёнида бу каби ноўрин ёзувларга йўл қўйилмаслиги керак.

10.2. Мослашувчан CSS яратиш

Юқорида веб-саҳифанинг кўринишини битта CSS файлни алмаштириш билан буткул ўзгартириб юбориш мумкинлигини таъкидлаб ўтдик. Умуман олганда CSS яратиш дизайнерларнинг иши ва уларнинг кўп йиллик тажрибаси веб-сайт яратишда иккита, нари борса учта рангни асос қилиб барча объектларни унга мутаносиб

тусда акслантириш күтилган самара беришни күрсатди. Бунда биринчи ранг асосий күзга ташланувчи объектиларга берилади, иккинчиси ортранг сифатида фойдаланилади. Күпинча ортранг оқдигича қөлади, аммо оқ күзни толиқтирувчи бўлганлиги боис уни ҳам ўзгартиришга талаб бўлмоқда. Учинчи ранг эса хусниятнинг кўриниши учун берилади. Қолган туслар, соялар, чегаравий чизиқлар шу рангларга мутаносиб равишда сал тўқ ёки сал оч тарзда берилиши мақсадга мувофиқ. Кейин хусниятлар баландлиги, тутмалар ўлчами ва майдонлар ҳажми экранда жойлашишига қараб кўрсатилиади.

Шундай вазиялар бўладики, сайт тайёр бўлгандан кейин танланган асосий ранг ёқмай қолиши, унинг ўрнига бошқаси мақбулдай туюлиши мумкин. Бундай вазиятларда CSS файлни бошқатдан янгидан яратиб чиқиш малол келиши аниқ. Ушбу муаммони ечиш учун «SASS»дан фойдаланилади.

Мукаммал дастурлашнинг асосий талабларидан бири, битта кодни қайтариб ёзмасликдир. HTML дастурлаш тили бўлмаганилиги учун бундай муаммога доим дуч келинади. Дастурлаш тиллари ёрдамида HTML да файл яратиб олиш кенг кўлланилади. Бундай ускуналар ичida Хамpton Катлин ва Натан Вейзенбаум томонидан яратилган **Haml** (HTML abstraction markup language) номли HTML тузиб берадиган жойлаштирув тили кенг оммалашган. Кейинги китобларимизда веб-дастурлаш тиллари билан кенг танишиб ўтамиз, шунинг учун «Haml»ни батафсил ўрганишга хожат йўқ. Лекин унинг асосида яратилган **Sass** (Syntactically Awesome Stylesheets — саҳифа шамойилининг бекиёс синтаксиси) ускунаси CSS файлларни мослашувчан тарзда яратишимиизда жуда асқотади.

SASSнинг синтаксиси бўйича янги қаторга ўтиш ва хатбошилар муҳим аҳамиятга эга, масалан:

SASS	CSS
<pre>div color: #333 align: center</pre>	<pre>div { color: #333; align: center; }</pre>

Бунда қаторнинг охирига «;» қўйиш ҳам талаб этилмайди. Аммо бу биз кўниккан CSS синтаксисидан кескин фарқ қиласди, агар қаторлар ўз ўрнида ажратилмаса ва қатор бошига бўш жойлар тўғри қўйилмаса, бундай файлни CSSга ўтказища хатолик юзага келади. Умуман олганда «Ruby» ёки «Python»да дастурлашни билмайдиган, ҳамда битта қаторга бир неча қоидаларни ёзишни маъқул кўрадиганлар учун SASSнинг синтаксиси ноқулайлик келтириб чиқаради. Шунинг учун ўзимиз ўрганганд CSS кўринишига айнан ўхшаш SCSS (Sassy CSS)дан фойдаланамиз.

SCSS файлни CSSга ўтказиш жуда осон, ўзингизнинг операцион тизимингизга мос ўтирувчини ўрнатасиз (sass-lang.com ёки sass-scss.ru сайтларининг биридан олишингиз мумкин) ва битта буйруқ бериш билан керакли CSS файлни ҳосил қиласиз. Агар бу ортиқча амалдай туюлса, тайёр онлайн ўтирувчилардан (масалан: codepen.io) фойдаланиш ҳам мумкин.

Қўйида SCSS файлни ёзиш қоидалари ҳамда уларнинг CSSга қандай тарзда ўтирилиши билан батафсил танишиб ўтамиз.

10.2.1. Ўзгарувчилар

Яратилиши лозим бўлган веб-сайт ёки ҳар қандай веб-тизимни олдин тасаввурга келтириб; қандай ранглар ишлатилади, ҳуснхатлар қанака бўлиши керак, майдон ва тутмаларнинг шакли, ундаги обьектларнинг жилоси қай тарзда намоён бўлиши лозимлигини аниқлаб олгандан кейин; уларни ифодаловчи ўзгарувчилар киритилади. Ўзгарувчиларнинг номлари «\$» белгиси билан бошланувчи келишилган идентификатор бўлиши керак, унда чизиқча (-) қатнашишига рухсат берилади. Ўзгарувчига олдин қиймат берилади, кейин шу қийматни ёзиш ўрнига ўзгарувчининг номи келтирилади. Масалан:

scss	css
\$main-color: darkblue; p { color: \$main-color; }	p { color: darkblue;

Қонуниятларни ёзиш мобайнида ўзгарувчининг қийматини алмаштириш мумкин. Айтайлик 200-қатордан кейин ички кичик бўлимчаларни жилолайдиган қоидалар ёзилган, унда «асосий ранг» бошқачароқ бўлиши лозим. Бундай вазиятда 200-қаторнинг ўзига қуийдагича ёзиб қўйиш кифоя: «\$main-color: magenta;».

10.2.2. Бириктириш

CSSда номланишлар уларнинг тадбиқ этиладиган соҳасига қараб турлича ёзилиши мумкинлиги билан танишиб ўтганмиз. [8.2.] Бундай мураккаб номланишларни ҳам SCSSда ифодалашнинг усули мавжуд. Бириктириладиган номланиш ташқисининг ичida жойлашади ва у CSSга ўтирилаётганда бўш жой билан қўшиб умумлаштириб ёзилади. Аммо табақалаш ва вазиятан синглашни ифодаловчи сўзлар қатнашганда номланишда икки нуктадан аввал бўш жой бўлмаслиги керак. Бундай вазиятда «&» белгиси ички номланишнинг бириктирилувчи бўлаги олдидан қўйилади:

SCSS	CSS
<pre>\$main-color: magenta; nav { color: \$main-color; li { display: inline-block; } ~ a { text-decoration: none; } &:hover {color: red; } }</pre>	<pre>nav { color: magenta; } nav li { display: inline-block; } nav ~ a { text-decoration: none; } nav:hover { color: red; }</pre>

10.2.3. Файлдан юклаб олиш

Баъзан шундай жилолаш қоидалари ёзиладики, уни кўпгина CSS файлнинг ичига қўшиш мақбул ҳисобланади. Ўниверсал шамойиллар, бежирим кўринишни ифода-

ловчи жозибаларни ҳар доим бирдай қонуният асосида табиқ этиш мақсадга мувофиқ келади. Бундай ҳолларда қисқа қонуниятлар түпламини алохіда файлга сақлад, уни керак жойда юклаб олиш лозим. Ихчам қонуниятлар ёзилган ушбу файлни номлашда биринчи ҳарғи «_» құллаш талаб этилади, албатта кенгайтмаси «.scss» тарзда бўлади. Бунақа номдаги файлларни дастурий таъминот орқали ўтирилаётганда эътиборсиз қолдирилади. Қачонки у бошқа «.scss» файлнинг ичидаги чақирилса, шундагина таркиби инобатга олинган ҳолда кўрсатилган жойга ўтирилиб «css» файл ҳосил қилинади. Масалан «`_main.scss`» номли файлнинг ичидаги «`html, body { margin: 0; padding: 10px; }`» деб ёзилган бўлсин. Уни қуийдагича чақириб, таркибидаги қоидаларни бошқа файлда табиқ этамиш:

SCSS	CSS
<pre>@import 'main'; body { background-color: aqua; }</pre>	<pre>html, body { margin: 0; padding: 10px; } body { background-color: aqua; }</pre>

Эътибор қаратиш лозимки, ихчам файлни юклаш учун «@import» калит сўзи қўлланиляпти ва ундан кейин қўштириноқ ичидаги файлнинг номи кегайтма ҳамда «_» сиз ёзиляпти. Ўгирувчининг ўзи керакли файлни номи ва кенгайтмаси бўйича топиб, таркибидаги қонуниятларни, ҳосил қилинаётган файл ичига «@import» жумласи ёзилган жойга муносиб равищдаги ерга қайта ишлаб ёзиб беради.

10.2.4. Функциядан фойдаланиш

Айрим хусусиятлар турли браузерларда ўзига ҳосном билан берилгандагина ишлайди. Юқорида «user-select» билан танишиб ўтган эдик. [9.8.] Бунда айни бир нарсани ҳар бир оммавий браузернинг талабини ино-

батга олган ҳолда қайта-қайта ёзиб чиқишимизга тұғри келади. Албатта бу бир томондан CSSнинг камчилигі бўлса, иккинчи томондан браузерларнинг нуқсони. Нима бўлганда ҳам ҳозирча битта қоидани бир нечта қилиб ёзишга мажбурмиз. Бундай вазиятларда SCSSдан фойдаланилса, код ёзиш самарадорлиги янада такомиллашади.

Бунинг учун ихтиёрий номда функция яратамиз ва унга ўзгарувчи сифатида киритилувчи қийматларни кўрсатишимиз мумкин. Функцияни эълон қилиш учун «@Mixin» («можаро» деган маънени англатади. Афтидан, бундай ноқулайликлар яратувчиларнинг орасида ҳам жиддий келишмовчиликлар келтириб чиқараётган бўлса керак) калит сўзидан фойдаланилади. Мисол:

SCSS	CSS
<pre>@Mixin userSelect(\$value) { color: maroon; user-select: \$value; -ms-user-select: \$value; -moz-user-select: \$value; -webkit-user-select:\$value; } div { @include userSelect(none); } p h1 { @include userSelect(none); }</pre>	<pre>div { color: maroon; user-select: none; -ms-user-select: none; -moz-user-select: none; -webkit-user-select: none; } p h1 { color: maroon; user-select: none; -ms-user-select: none; -moz-user-select: none; -webkit-user-select: none; }</pre>

10.2.5. Мерос олиш

Кўп ҳолларда битта объект учун тайинланган қоидалардан андоза олиб, бошқа номланишли шартлар мажмуасини яратишга тұғри келади. Бунда номланишнинг ўзини чақириб қўйиш кифоя:

SCSS	CSS
<pre>.title { border: 1px solid #ccc; border-radius: 3px; padding: 2px 10px; color: #333; } h1 { @extend .title; font-size: 18px; } h2 { @extend .title; font-size: 16px; color: maroon; }</pre>	<pre>.title, h1, h2 { border: 1px solid #ccc; border-radius: 3px; padding: 2px 10px; color: #333; } h1 { font-size: 18px; } h2 { font-size: 16px; color: maroon; }</pre>

Бир қараганда SCSS да ёзилган код CSS га ўтирилган дагидан кўпдай, мерос олиш самарали эмасдай туюлади. Аммо жуда катта ҳажмли файлларни таҳрирлашда уларнинг ичидан андоза олинаётган номланишни топиб ўзгартириш анча мушкуллик туғдиради. Айниқса юкландиган ихчам файлни [10.1.3] кейинчалик ўзгартиравериш мақсадга мувофиқ эмас, қолаверса, ундаги шартлар янги «CSS» учун бошқача тадбиқ этилиши мумкин. Умумий қоидалар қандай номланишларга тааллуқли бўлиш шартлигини аввалдан тўлиқ билиб бўлмаганилиги сабабли «мерослаш»дан фойдаланиш самара беришини амалда синаса бўлади.

10.2.6. Математик амаллар

SCSSнинг энг фойдали жиҳати математик аммалларни қўллай олишидир. Ўлчамларни олдиндан киритилган ўзгарувчига нисбатан ҳисоблаб топиш тушунарли ҳол. Аммо ҳисоб-китоб содда тўрт математик амал ёрдамида нафақат сонли микдорлар, балки ранглар учун ҳам тадбиқ этилади. Бунда рангни ўн олтилик саноқ тизимидағи олти хонали сон орқали ифодаланилиши

инобатга олинган. Қуйида h5 тегида ёзилган, объект-нинг марказида чиқиши назарда тутилган мавзуларни тутмасифат кўринишида тасвирловчи жилолашга мисол келтирамиз:

SCSS	\$button-height: 24px; \$btnColor: #eee; \$caption-color: #333; h5[align='center'] { height: \$button-height; width: \$button-height * 5; font-size: \$button-height * 12 / 10; margin: 6px; padding: \$button-height / 6; font-weight: bolder; cursor: pointer; background-color: \$btnColor; color: \$caption-color; border-radius: 6px; box-shadow: 2px 2px 3px \$btnColor - #666, inset 2px 2px 3px \$btnColor - #333; &:active { color: \$caption-color + #222; box-shadow: inset 2px 2px 3px \$btnColor - #666, 1px 1px 2px \$btnColor - #333; } }
SS	/* Фақат «align="center"» аломатли h5 теглар учун */ h5[align='center'] { height: 24px; /* Баландлиги, ўзгарувчан */ width: 120px; /* Эни, бўйидан беш баробар узун */ font-size: 28.8px; /* Хуснхат баландлиги,h5 ўлчамига мос */ margin: 6px; /* Ташки ҳошия, қўшни объектдан ажратади */ padding: 4px; /* Ички ҳошия, матн чегарадан узоқлашади */ font-weight: bolder; /* Калин ёзув */ cursor: pointer; /* Сичқонча курсатгичи, ишоратницидай */ background-color: #eee; /* Объект ортранги */ color: #333; /* Матн ранги */ border-radius: 6px; /* Объект чегарасининг қиррасини ёйсимон қилиш */

CSS	<pre> box-shadow: 2px 2px 3px #888888, inset 2px 2px 3px #bbbbbb; /* Соялар */ } /* Босиб түрилгән ҳолат, түгма тасаввурини беради */ h5[align='center']:active { color: #555555; /* Матн ранги, сал очаради */ box-shadow: inset 2px 2px 3px #888888, 1px 1px 2px #bbbbbb; } /* Соялар алмашади, түгма босилғандай күриншни учун */ </pre>
-----	---

HTML	<pre> <-- Түгмасыфат мавзу --> <h5 align="center">HTML</h5> <h5 align="center">CSS</h5> <-- Оддий мавзу --> <h5>Мұкаммал құлланма</h5> </pre>	Натижা	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> HTML </div> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> CSS </div> <p>Мұкаммал құлланма</p>
------	---	--------	---

10.3. Модал ойна

Модал ойна шундай блокли тегнинг аксланишики, намоён бўлувчи объект ва унинг ичидагилардан ташқаридағи барча элементлар нофаол тарзда бўлади. Фойдаланувчи ягона фаол ойнадаги талаб этилган амалларни бажаришини, бошқа элементлар «қотган ҳолда» кутиб туришини, тақозо этади. Бу баъзи қийматларни киритишини талаб этиб, қолган жараёнларни уларга боғлиқ равишда бажариш лозим бўлганда фойдаланувчи томонидан кутилмаган хатоликка йўл қўйилишини олдини олади. Демак модал ойна акслангандা, у ўз ички элементлари билан бирга саҳифадаги ягона фаол объект бўлади.

Аслида HTML5 ва CSS3 имкониятлари билан модал ойнани яратиш жуда осон. Бунинг учун `<dialog>` теги ва `::::backdrop` вазият синфи мавжуд. Аммо буларни Хром ва Операдан бошқа браузерлар қўлламагани боис, мураккаброқ усулдан фойдаланишга мажбурмиз.

Бутун саҳифани тўлиқ қоплайдиган блокли тег яратамиз ва уни энг усти қатламда жойлашишини таъминлаймиз. Керак бўлганда ушбу объектни кўрсатамиз

ва ундаги мурожаат ойнасини акслантирамиз, қолган ҳолларда яширин ҳолатда бўлади. У бошқа қатламда намоён бўлаётгани учун саҳифадаги бошқа элементалар жойлашишига таъсир кўрсатмайди. Аввал модал ойнани чақирувчи ёрлик қўяйлик, уни босганда бошқа саҳифага ўтмайди, фақатгина ишоратнинг охирги қисми ўзгаради, яни «#»дан сўнг ёрлик номи қўшилади.

HTML кодини қўйида келтирилгани каби қисқача ифодалаш мумкин:

```
<a href="#showModal">Модал ойна</a>
<div id="showModal" class="modalQatlam">
    <div>
        <a href="#yop" title="Ёниш" class="yopish">x</a>
        <h2>Фойдаланувчи билан мулоқот</h2>
        <p>
            Бу ерда танловлар ва мурожаатлар
            акслантирилиши мумкин
        </p>
    </div>
</div>
```

Энди ҳар бир элементнинг қандай тарзда аксланишини CSS ёрдамида аниқлаймиз:

```
.modalQatlam {
    position: fixed; /* Бошқа қатламда координатаси аниқ кўрсатилган объект */
    top: 0;
    right: 0;
    bottom: 0; /* тўрт тарафга тўлиқ ёйилган, саҳифани тўла эгаллаган */
    left: 0;
    z-index: 999; /* Энг юқори қатламда (бошқа «z-index»лар 999дан кичик) */
    background: rgba(222, 222, 222, 0.6); /* Орттасвири хира куранг */
    opacity: 0; /* Кўринмайдиган даражада шаффоф */
    transition: opacity 0.5s ease-in; /* Шаффофлик аста алмашади */
    pointer-events: none; /* Сичқонча босишларини бекор қиласади */
}
.modalQatlam:target { /* «Модал ойна» босилгандагина ишлайди */
    opacity: 1; /* Кўринмайдиган даражага келтиради */
    pointer-events: auto; /* Сичқонча босишлари инобатга олинади */
}
```

```

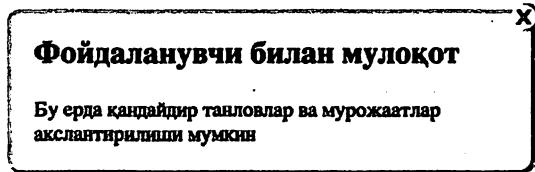
.modalQatlam > div { /* Фақат модал қатламнинг ичидаги «div» учун */
    width: 360px;           /* Мурожаат ойнасиниг эни */
    position: relative;     /* «modalQatlam»га нисбатан жойлашып */
    margin: calc(50% - 100px) auto; /* марказда жойлашади */
    padding: 5px 20px;       /* Ички ёзувларнинг чегарадан узоқлаштиради */
    border-radius: 10px;     /* Ойна құрраларини ёйсилген қиласы */
    background: #dadada;   /* Ортасындағы күлранг */
    box-shadow: 2px 2px 3px #000, inset 2px 2px 3px gray;
    /* Ойна бүртіб чиққандай тасаввурни берши учун иккі хил соя құйылған */
}

.yopish { /* Ойнани ёпиши учун нисонча */
    background: #dbbdb;      /* Нисонча ортранги */
    font-family: Consolas;   /* «x» қарфмас, ёпиши тасвиридай күрінади */
    color: #666;             /* «x»нинг ранги */
    line-height: 20px;        /* «x»нинг баландлығы, эніга мутаносиб бўлади */
    width: 20px;              /* Нисонча эни, ёзув ҳажми инобатга олинади */
    font-size: 24px;          /* «x» көттәрөк күрниши учун */
    position: absolute;       /* жойлашып ўрнини аниқ күрсатиш учун */
    right: -8px;              /* ўнгга чиқиб түради */
    top: -4px;                /* төпадан ҳам чиқиб түради */
    text-align: center;        /* «x» ўртада жойлашади */
    text-decoration: none;     /* остики чизиқ акслантирилмайди */
    font-weight: bold;         /* қалин ёзув */
    border-radius: 12px;       /* доирасибат күрнишини таъминлайди */
    box-shadow: 2px 0 3px #000; /* сояси ойнаникига мутаносиб */
}

.yopish:hover { background: #eee; } /* сичқонча борганини билдириш учун */

```

Ушбу кодлар ўз ўрнига қўйиб бажарилса, сахифада пайдо бўлган «Модал ойна» ёзуви босилганда экран марказида куйидагича тасвир пайдо бўлади:



Албатта таъбга қараб ҳар ким ўзи истагандай шаклшамойил бериш мумкин. Ойнанинг баландлиги олдиндан аниқ бўлмагани боис, тик равишда марказга чиқариш ҳар доим ҳам муваффақиятли амалга ошмайди. Шунинг учун юқорига нисбатан аниқ масофада жойлаш маъқул топилади, шунда calc функциясини ишлатишга ҳожат қолмайди.

10.4. Қисқа изоҳ

Кўп ҳолларда берилган матн акслантирилиши лозим бўлган майдонга сифмайди. Қолаверса майдоннинг ўзи ҳам турли мониторлар учун турлича ўлчамда бўлади. Агар матн учун бир қатор жой ажратилган бўлса, масалан, янгиликлар учун мўлжалланган сайтдаги мавзулар рўйхати сингари, унинг йўли осон. Қисча мавзулар тўлиқлигича кўринаверади, узўн сарлавҳалар эса майдон энита еттанди кесилиб, тўлиқ эмаслиги кўриниши учун уч нуқта кўйилади. Бунинг учун CSS қўйидагича ёзилади:

```
.izoh {  
    width: 30%; /* эни қанча бўлишини аниқ чегараланади */  
    height: 20px; /* баландлигини ёзув ўлчамига мос кўрсатиш лозим */  
    font-size: 16px; /* ёзув рангини таъбиға қараб кўрсатиш мумкин */  
    color: darkblue; /* ягона сатрдан иборат бўлиши белгиланади */  
    white-space: nowrap; /* факт ягона сатрдан иборат бўлиши белгиланади */  
    overflow: hidden; /* сизмаган ёзув қисми кўринмаслиги таъминланади */  
    text-overflow: ellipsis; /* ёзув кесиласа, охирiga уч нуқта қўйилади */  
}
```

Бунда бир неча хусусиятларнинг мутаносиб вазифа бажаришини инобатта олиш лозим. Агар эни аниқ кўрсатилмаса, ёзув ҳажмита қараб у ўзгаргани боис, бејиримлик йўқолиб, керакли жойда кесилмаслиги мумкин. Одатда сарлавҳа тўлиқ ёзилиб, унга бериладиган изоҳлар учун бундай қисқартиришлар кенг қўлланилади. Агар кейинги қаторга тушишни таъқиқланмаса, қандай узунликдаги сўз бошқа сатрга ўтгани аниқ бўлмаганилиги боис, кесилган қаторлар ўнг тарафдан текис жойлашмайди. Яна бир муҳим жиҳати «overflow: hidden;» қиймати берилмаса, «text-overflow» ўз вазифасини бажармайди.

Тасаввур мукаммалроқ бўлиши учун аниқ мисол кўриб ўтайлик:

HTML <pre> <h2> HTML</h2> <p>Веб-сақиғаны жиҳозлашга йұналтирилған тил.</p> <h2> CSS </h2> <p>HTML элементтеринің жиһолаша мүлжалланған қонунияттар мажмаси.</p> <h2> XML </h2> <p>Маълумоттарни ғақат қатың теглар ёрдамида бериладиган шакли.</p> <h2> SVG </h2> <p>Махсус теглар ёрдамида тасвир чизишке йұналтирилған тил.</p></pre>
CSS <pre> h2 { width: 60px; height: 20px; padding: 2px 4px; /* Ички хошиялар қалындығы, төлең-наст ва ёңіларидан */ text-align: center; /* Ички матн сатрларын үртаса қойылацади */ float: left; /* Чапта текисланади, үндегде элементтерниң мүмкін */ background: #dedede; cursor: default; /* Күрсектігін шаклини одатдайды */ } p { width: 380px; height: 20px; padding: 2px 4px; font-size: 16px; /* Хүсніхаттың үлчамы */ background: #eee; /* Ортранги */ color: darkblue; /* Ранги */ white-space: nowrap; /* Ғақаты биттіңде қаторға ёзилади */ overflow: hidden; /* Сигмаган қисмын күрінмайды */ text-overflow: ellipsis; /* Матн кесілса, уч нүктә қўйылади */ }</pre>
НАТИКА <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;"> HTML Веб-сақиғаны жиҳозлашга йұналтирилған тил. </div> <div style="width: 30%;"> CSS HTML әлемнің элементтеринің жиһолаша мүлжалланған т... </div> <div style="width: 30%;"> XML Маълумоттарни ғақат қатың теглар ёрдамида... </div> <div style="width: 30%;"> SVG Махсус теглар ёрдамида тасвир чизишке йұна... </div> </div>

10.5. Ҳаракатланувчи меню

Веб-сайт яратилар экан биринчи ўринда ундаги маълумотлар бўлса, иккинчи ўринда унинг қандай жозибада намойиш этиш муҳим. Саҳифадаги обьектлар вақтга ёки фойдаланувчининг таниловига қараб ўзгарар экан, ушбу жараён ўзига ҳос ҳаракатланиш орқали амалга оширилса, муштарийнинг ўзига жалб этиш учун бу яхши омил сифатида хизмат қиласди. Ҳаракатлантириш албатта маҳсус ҳисоб-китоблар орқали бажарилади, бунинг учун эса дастурий татъминот ҳамда қўшимча скрипталар талаб этилиши турган гап. Аммо CSSнинг стандарт ҳаракатлантириш воситалари ҳамда HTMLдаги ҳодисаларни бирбирига боғлаш орқали бир қанча анимацияларни яратишимиш мумкин.

Юқорида ўрганган билимларимиз асосида очилиб-ёпилувчи изоҳли меню яратамиз.

HTML кодини қуйида келтирилгани каби қисқача ифодалаш мумкин:

```
<aside id="chapTomon"> <!-- Сарлавҳаларни ўз ичига олган майдон -->
    <details>           <!-- Биринчи яширин изоҳли сарлавҳа -->
        <summary>Мавзу 1</summary> <!-- Биринчи сарлавҳа номи -->
        <p>
            Уч<br>қатор<br>изоҳ <!-- Изоҳ аниқ баландликка эга -->
        </p>
    </details>
    <details>           <!-- Иккинчи яширин изоҳли сарлавҳа -->
        <summary>Мавзу 2</summary> <!-- Иккинчи сарлавҳа номи -->
        <p> Қандайдир изоҳ </p>
    </details>
</aside>
```

Энди ҳар бир элементнинг қандай тарзда аксланишини CSS ёрдамида аниқлаймиз:

```

#chapTomon { /* изохли сарлавхалар чиқши майдони */
    width: 185px; /* майдон эни, сарлавхә энидан көттөрөк */
    border: 1px solid #dedede; /* майдон чегарасын күлгөнгө узлуксиз чизиқ */
    border-radius: 6px; /* чегара қирралари ёйсімден қилинган*/
    float: left; /* майдон чапда, ўнгдан объекттер чиқши мүмкін*/
}
details { /* изохли сарлавхә құйиши теги */
    border-width: 1px; /* чегара қалынлігі, анық ұлчамда күрнешіши үчүн */
    background: #efefef; /* ортрангы, оч күлгөнгө */
    padding: 2px; /* атрофидаги ичкі ҳисия */
    font-size: 18px; /* хисніхат ұлчами */
    font-family: Arial; /* хисніхат номи */
    line-height: 20px !important; /* қатар баландлігі 20 пиксель бұлшып! */
    min-height: 20px; /* әнд кичик баландлігі */
    animation: yopil .3s ease-in; /* «yopil» номлы ҳаракат 0.3 сонияда бажарылады */
}
details[open] { /* изох очилған ҳолати */
    max-height: 100px; /* әнд баланд қиймати */
    overflow: hidden; /* очилаётгандың сияғынан өзүвлөр күрнешмейді */
    animation: ochil .8s ease-out; /* «ochil» 0.8 сонияда бажарылады */
}
details summary { /* «details» ичидеги «summary» теглары, сарлавхалар */
    background: #dedede; /* ортрангы изохдан ажраб түриши үчүн түйкөрөк */
    border-radius: 3px; /* ёйсімден бүрчактың түгел шекаралық келиши үчүн */
    box-shadow: inset 2px 2px 4px grey, 2px 2px 4px black; /* бүрттіпірбіл чиқарыш */
    padding: 2px 2px 2px 12px; /* өзүв сал ұнгроқдан бойланады */
    font-weight: bold; /* сарлавхә қалын өзүвде күрнешады */
    width: 160px; /* майдон энидан (хисия ва чегара билан) кичик */
    cursor: pointer; /* сичқонча күрсатғычи (күл күрчатғычи нисионча) */
}
details p { /* «details» ичидеги «p»теглары, изох өзүві */
    margin: 10px 24px; /* сарлавхага нисбеттің ичкарида бұлшышини үчүн */
    color: #333; /* өзүв ранги (ажрат түриши үчүн) */
}
@keyframes ochil { /* «ochil» номлы ҳаракатланышын таъминлаш */
    from { height: 20px; } /* бойланғыч вазият, баландлігі сарлавханықи */
    to { height: 100px; } /* якуний вазият, әнд жоғори баландлік */
}
@keyframes yopil { /* «yopil» номлы ҳаракатланышын таъминлаш */
    from { height: 100px; } /* бойланғыч вазият, әнд жоғори баландлік */
    to { height: 20px; } /* якуний вазият, баландлігі сарлавханықи */
}

```

► Mavzu 1

Уч
қатор
изоҳ

► Mavzu 2

10.6. Қалқиб чиқувчи ойна

HTMLда ҳар бир объект яратувчи тегнинг «title» номли аломати мавжуд, унга берилган матн сичқонча ўша объект устига боргандা изоҳ сифатида кўринади. Ушбу фойдали қисқа ёзувни исталганча жилолаш, кўриниш вақтини ва жойлашиш ўрнини ўзгартириш афсуски мумкин эмас. Уни браузернинг ўзи аниқлайди. Аммо унинг ўрнини боса оладиган қалқиб чиқувчи ойнани ўзимиз яратишими мумкин.

Аввал бир қизиқарли ва жуда фойдали CSSнинг имкониятини таҳлил қилиб ўтайлик. Ҳар бир объект яратилишидан олдин ва кейин унинг танасига бириктирилган ҳолда пайдо бўлувчи, алоҳида-алоҳида иккита қўшимча элемент шакллантириш мумкин. Юқорида ушбу имкониятдан бўлим номларининг олдига рақам қўйишда фойдаланган эдик. [9.9.] Умуман олганда блокли теглар учун тўғри тўртбурчак шаклидаги алоҳида майдон ҳосил қилиш имконияти мавжуд. Ҳар бир блокли тег тўғри тўртбурчак шаклини эгаллашини инобатга олиб, унинг периметридан учбурчак ясаш учун фойдаланамиз. Бунинг учун чегаравий чизиқни қалинроқ қиласиз ва ўлчамини нолга tenglashтирамиз. Бунда чегара ўлчамига икки баробар катталиқдаги тўғри тўртбурчак ҳосил бўлади. Биламизки, тўрт томондаги чегаравий чизиқларни тўрт хил тусда чизиш мумкин, бу билан тўртта учлари бир нуқтада бирлашган учбурчак ҳосил бўлади:

```


Албатта ҳар бир томон учун қалинликни ҳам турлича бериш мүмкін, бу билан исталған шаклдаги учбұрчак ҳосил қилиш мүмкін. Қолған утасининг рангини шаффофф қилиб қўйилса, керакли ягона учбұрчак кўринарли бўлади. Унинг атроф хусусиятлари* ёрдамида ихтиёрий нуқтага қўчириш мүмкін. Аввал керакли учбұрчак ҳосил килувчи код ёзиб келтирайлик:



```

#uchburchak {
 width: 0;
 height: 0;
 border-top: 80px solid #ddd;
 border-left: 60px solid #ddd;
 border-right: 30px solid #ddd;
 border-bottom: 40px solid #333;
}

```



Қалқиб чиқувчи ойнадаги матн қайси обьект учунлигини кўрсатиш мақсадида «мил» қўйиб қўйиш анъанага кирган. Блокли тегларимиз тўгри тўртбұрчак шаклидаги обьект ҳосил қиласар экан, унинг керакли жойига учбұрчакни шундай жойлаштириш лозимки, обьектга кўрсатувчи «мил»ли хабар берувчи матнли шакл тасавурини берсин:



196


```

Албатта унинг сояси ва чегаравий чизифига эътибор бериш лозим. Аммо биз ҳосил қилган учбурчак аслида чегаранинг ўзи, унинг ички ранги мавжуд эмас. Ўлчами йўқ объектнинг сояси ҳам бўлмайди. Шунинг учун «мил»ни соя тушмайдиган тарафга қўйиш маъқул. Кўрсатувчи «мил»нинг ва унинг чегарасининг рангини объектнигида ўхшатиш учун иккита учбурчак чизамиз. Биринчиси сал каттароқ бўлиб, чегара рангида бўлади. Иккинчиси унинг устига объект ортрангида чизилади. Шунда тагида қолиб кетган катта учбурчак устидагисининг чегаравий чизиги сифатида кўринади. Фақат аввал чегара рангидаги каттароқ учбурчакни «:before» табақасида, сўнг ички уббурчакни «:after» табақасида қоидаларини ёзамиз ва жойлашиш нуқтасини мос равища кўрсатамиз.

Айтайлик, бизнинг қалқиб чиқувчи ойнамиз «izoh» синфиға тааллуқли «р»тегининг объектлари учун мўлжалланган бўлсин. Ҳозирча фақат CSS имкониятларидан келиб чиқиб, «ойна»ни кўзгалмас жойда яшириш ёки кўринарли қилиш билан шуғулланамиз. Кейинги JavaScriptга бағишлиган китобда «ойна»ни сичқонча ҳаракатига қараб аста кўчириш, мили ва матнини ўзгартириш, қандай амалга оширилиши билан батафсил танишамиз. Аввало CSS қоидалари асосида мили юқорига қараган ва сояси пастга тушган қалқиб чиқувчи ойнани кўринмас ҳолда яратамиз:

```
p.izoh { /* қалқувчи ойна фақат <p class="izoh">лар учун */
  position: absolute; /* юқори қатламда жойлашади */
  visibility: hidden; /* дастлаб кўринмайдиган ҳолатда бўлади */
  background-color: #eee; /* ортранги */
  border: 3px solid gray; /* чегаравий чизиқлари, қалин ва тўқроқ */
  border-radius: 5px; /* қирралари ёйсимон */
  padding: 4px 8px; /* матн чегаравий чизиқдан ичкарироқда */
  font-weight: bold; /* қалин ёзув */
  top: 50px; /* юқорига нисбатан 50 пиксел пастда */
  left: 18px; /* чапга нисбатан 18 пиксел ўнгроқда */
  box-shadow: -3px 3px 5px #000;
  /* соя чап пастда, 5 пикселгача энлайди */}
```

```

p.izoh::before { /* «р»дан олдин чизилади */
  content: '';
  position: absolute; /* жойлашув ўрнини ўзгартириш учун */
  top: -15px; /* «р»га нисбатан 15 пиксел юқорида */
  left: 11px; /* «р»нинг бошлангич нуқтасидан ичкарироқда */
  width: 0; height: 0; /* ўзининг ўлчами йўқ */
  /* тенадаги учбуручак кўринмас, ўлчами аҳамиятсиз */
  border-top: transparent;
  /* ўнгдаги учбуручак кўринмас */
  border-right: 15px solid transparent;
  /* чандаги учбуручак кўринмас */
  border-left: 15px solid transparent;
  /* настдаги учбуручак кўринади */
  border-bottom: 15px solid gray;
}
p.izoh::after { /* кейинги учбуручак ҳам юқоридагидай ясалади */
  content: '';
  position: absolute;
  top: -10px;
  left: 14px;
  margin-left: 2px;
  width: 0;
  height: 0;
  background-color: transparent;
  border-top: transparent;
  border-right: 10px solid transparent;
  border-left: 10px solid transparent;
  border-bottom: 10px solid #eee;
}

```

Қалқиб чиқувчи ойна билан браузернинг ўзидағи «title»га берилган изоҳни кўрсатишини кескин фарқлаш учун ҳар иккаласи ҳам кўринадиган мисол тузайлик. Харитали расм қўйиш [5.1.] мавзусида расм учун майдон чизиб сичқонча орқали расм устида кўрсаттични юргизганда мос изоҳ чиқишига доир мисол кўрган эдик. Энди кичик изоҳчага мос равишда, у билан бирга кенгайтирилган матнни ўз ичига олган қалқиб чиқувчи ойна ҳам кўринсин.

Алишер Навоийнинг бир ажойиб тўртлигида «жон» сўзидағи ҳарф ва белгиларни баҳшида этиши ҳақида ёзил-

ган. Агар араб алифбоси асосидаги эски ўзбек ёзуви тасвур қилинмаса, шеърда нима назарда тутилганлигини тушуниш мушкул. Шунинг учун араб ҳарфлари ёрдамида «жон» деган сўзни ёзамиз ва уни расм сифатида сақлаб оламиз. Ўлчами 60x40 пискел бўлган тасвир ҳосил бўлди. Энди «тар» теги ёрдамида унданда ҳар бир ҳарф ва нуқталарни қаерда жойлашганини белгилаб чиқамиз. Ҳар бир майдонни ифодаловчи «area» тегининг «title»ига мос белги номини берамиз ва ундан кейин тўртликнинг тегишли қаторини «`p class="izoh"`» теги ичида келтириб ўтамиз. Ўзимизга тушунарли бўлган қуийдагича код ёзилади:

```
<map name="harflar">
  <area shape="poly" title="Жим" coords="36,22,48,16,58,24,59,30,36,30">
  <p class="izoh">«ЖОН»имдаги «жим» икки долинита фидо.</p>
  <area shape="rect" coords="28,2,36,32" title="Алиф">
  <p class="izoh">Андин сўнг «алиф» тоза ниҳолинита фидо.</p>
  <area shape="poly" title="Нун" coords="4,20,18,30,26,18,28,39,3,39">
  <p class="izoh">«Нун»и доги анбарин ҳилолинита фидо.</p>
  <area shape="circle" coords="15,12,4" title="Нуқта">
  <p class="izoh">Қолғон икки нуқта икки холинита фидо.</p>
  <area shape="circle" coords="45,34,4" title="Нуқта">
  <p class="izoh">Қолғон икки нуқта икки холинита фидо.</p>
</map>

```

«Жон» деган сўздан ташкил топган расмдаги «жим», «алиф», «нун» ҳарфлари ва яна иккита нуқтани белгилаб, уларга мос Навоий мисраларини керакли тартибда жойлаштириб чиқилди.

Энди сичқончанинг кўрсатгичи қайси «area» устига борган бўлса, ундан кейин жойлашган «`p class="izoh"`» элементи кўринадиган бўлишини таъминлашимиз керак. Бунинг учун «қўшни тег»га [8.2.] мурожаат қилиш синтаксисидан фойдаланиб, CSS файлга қуийдагиларни қўшимча қилиш кифоя:

```
area[title="Жим"]:hover + p { visibility: visible; }
area[title="Алиф"]:hover + p { visibility: visible; }
area[title="Нун"]:hover + p { visibility: visible; }
area[title="Нуқта"]:hover + p { visibility: visible; }
```

Тасаввур түлиқ бўлиши учун ва ушбу мавзуда баён этилган имкониятларни тўғри тушуниб етиш мақсадида келтирилган мисолларни амалда синаб, ишлатиб кўриш тавсия этилади. Ягона ҳол учун, сичқонча расмдаги «алиф» ҳарфи устига борганда қуидагича кўриниш намоён бўлади:



10.7. Йиғиладиган майдон

Веб-сайтнинг бош саҳифасида имкони борича барча йўналишини қамраб оладиган тарзда кўп маълумот келтирилиши керак. Лекин инсон эътибори ҳамма нарсага тенг тушмаслигини инобатта олиб, ортиқча маълумотларни кўрсатишдан тийилиш мақсадга мувофиқдир. Масалан, сайтларда одатда қўйиладиган «биз ҳақимизда» ёки «биз билан алоқа» каби бўлимчаларда ёзиладиган манзилларни кўпчиликка аҳамияти йўқ. Лекин мурожаат қилмоқчилар бўлса, албатта улар учун илк саҳифадаёқ имконият яратувчи ойна қўйилиши лозим. Бу каби ахборотларни йиғиладиган ойна ёки майдон ичида келтирилса, қизиқсан фойдаланувчи номи бўйича ўзига керакли бўлганини очиб, ундаги маълумотни кўриш имконияти яратиласди. Кўп холларда тажрибасиз дастурловчилар юқори ёки ён тарафга жойлашган меню ичида шу каби ахборотларни кўрсатишида, бу билан менюнинг умумий қоидаси бузилади, аслида у ерларда алоҳида саҳифада бошқа маълумотларни кўрсатувчи ишоратлардан ташкил топган тутмалар рўйхати бўлиши лозим.

Ҳозирча биз фақаттина HTML ва CSS имкониятларидан келиб чиқиб, ойна тасаввурини берадиган ёйиладиган майдонча яратишни кўриб чиқамиз. У «checkbox» танланганда, яъни дарчага белги қўйилганда секин ёйилувчи ва аксинча секин йиғилувчи «div» объектидан иборат бўла-

ди. (Кейинги китобда JavaScript ёрдамида бир қанча ходисаларга боғланган ҳолда турлича жилолашларни ҳам кўриб ўтамиз.)

Ойнанинг тепа қисмида графикли операцион тизимларда кўнинканимиз каби форманинг номи ва нишончалари жойлашади. Шу билан бирга ҳаққоний кўриниши учун орқа ранги тусланувчан ҳамда чегаравий қирралари ёйсимион тарзда бўлади. Унинг учун CSS куйидагича ёзилади:

```
.maydon { /* умумий ташки майдон */
    width: 300px; /* ойнанинг эни, бўйи таркибига қарраб ўзгаради */
    margin: 4px; /* қўйини объектлардан ажраб туриши учун */
    position: relative; /* ички жойлашувлар шунга нисбатан олиншии учун */
}
#bosh { /* ойнинг тепа сарлавҳа қисми */
    height: 24px; /* бўйи бўйи қатор ёзув ва нишончи ўлчамига яос */
    cursor: pointer; /* кўрчатигич кўриниши ўзгартирилган */
    background-image: linear-gradient(to top, #666, #eee); /* тусланувчан */
    border: 1px solid grey; /* чегара қалинлиги */
    border-top-left-radius: 12px; /* тепа қисмида қирралар ёйсимон */
    border-top-right-radius: 12px;
    box-shadow: 3px 3px 5px #333; /* соя ўнг пасига йўналган */
}
#tanla { /* танлаш дарчаси */
    position: absolute; /* жойлашиши майдонга нисбатан ўзгартириши учун */
    top: 3px; /* «bosh» қисми ичида чиқиш учун */
    left: 8px;
}
#mavzu { /* Сарлавҳа */
    position: absolute; /* белгиланувчи дарча ёнидан чиқиш учун */
    float: left;
    top: -12px;
    left: 30px;
    font-weight: bolder; /* қалин ёзув */
    user-select: none; /* сичқонча ёрдамида матн белгиланмаслиги учун */
    -moz-user-select: none;
    -ms-user-select: none;
    -webkit-user-select: none;
}
#tanla:checked + div { /* белгиланувчи дарчадан кейинги ойна */
    height: 200px; /* бошлангич баландлиги */
    animation: ochil .7s ease-out; /* белгиланганда 700 миллисониядада ёйилади */
}
#tanla:not(:checked) + div {
    animation: yopil .7s ease-out; /* белги олинганда 0,7 сонияда йигилади */
    height: 0;
}
#oyna { /* йигилиб-ёйилувчи ойна */
    margin-top: -3px;
    border: 1px solid grey;
    height: 200px;
}
```

```

background-image: repeating-radial-gradient(circle at center, #999, #eee 300px);
box-shadow: 3px 3px 5px #333, inset 1px 2px 1px #333;
border-bottom-left-radius: 6px;
border-bottom-right-radius: 6px;
overflow: hidden; /* ойна киңраяётіганды сегмаган матн күрингімдай */
}
@keyframes ochil { /* «ochil» номлы ұфракатланишини таъминлаш */
  from { height: 0; } /* башланғыч вазият, баланделуғы нол */
  to { height: 200px; } /* якуний вазият, әнг төкөри баланделуқ */
}
@keyframes yopil { /* «yopil» номлы ұфракатланишини таъминлаш */
  from { height: 200px; } /* башланғыч вазият, әнг төкөри баланделуқ */
  to { height: 0; } /* якуний вазият, баланделуғы мавжуд әмас*/
}

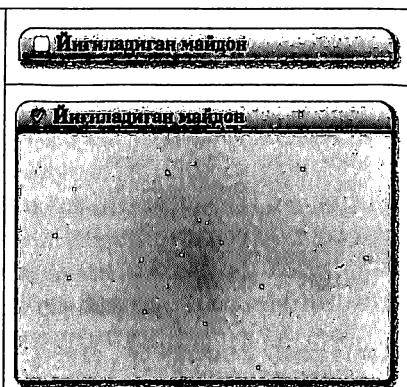
```

Агар «.maydon»га «position: relative;» қиймати берилмаса, унинг ичида жойлашган элементлар учун «position: absolute;» хусусияти саҳифага нисбатан ҳисобланар әди. Майдон саҳифаның қаерида жойлашишини қаттый аниқланмаган вазифаларда, айниқса, күчувчан объектлар яратында ушбу хусусиятта аҳамият қаратиш шарт. Яна бир мұхим жиҳати, құшни тегларни мутаносиб жойлаштириш ва сояларини умумлаштириш. HTML коди CSSда ёзилғандан анча ихчам бўлади:

```

<div class="maydon">
  <div id="bosh"></div>
  <p id="mavzu">
    Йигиладиган майдон
  </p>
  <input id="tanla"
    type="checkbox">
  <div id="oyna"></div>
</div>

```

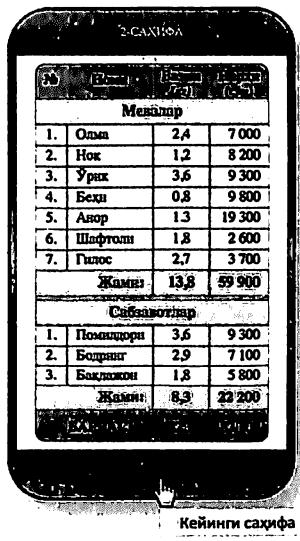


Бир марта ёзилған CSS қоидалари асосида бир қанча HTML объектлари яратылғанын көрсетет. Фақат унда қоидаларни «id»га боғлаш тавсия этилмайды.

10.8. Мұйжаз ҳисоботлар

Дастурнинг эң мухим қисми у чиқариб берадиган ҳисоботлардадир. Бириңчи ўринда чиқаёттан маълумотлар түғрилиги инобатта олинса, иккинчидан қандай кўринишда чиқаётганига аҳамият қаратилади. Шундай экан, ҳисобот кўринишига алоҳида эътибор беришимизга тўғри келади.

Хозирги замонавий технология юксалаётган пайтда, фойдаланувчилар компьютер ва лаптоплардан ҳам кўра смартфон ва планшетларни ишлатишни маъқул кўришмоқда. Сабаби улар ихчам, куввати ҳам кўпроқча етади, ҳамда ортиқча мураккаб дастурий таъминотлардан ҳоли. Шунга мос равищда тақдим этиладиган ҳисоботларнинг кўриниши ҳам кичик ва бежирим бўлиши талаб этилмоқда. Юқоридагиларни инобатта олиб, айнан уяли курилма кўринишидаги кўйидаги расмда кўрсатилгандек ҳисобот тайёрлаш коди билан танишиб чиқамиз.



Кейинги саҳифа

Битта саҳифада кўп маълумот кўрсатиш тавсия этилмайди. Энг керакли ахборотларни жамлаб уларнинг тоифаси бўйича бир неча саҳифага бўлиб чиқариш маъқул. Фойдаланувчи ўзига кераклисини қараб, лозим топса чоп этиб оләверади.

Масала мукаммалроқ бўлиши учун уни сал мураккаблаштириб, юқоридаги кўринишдаги жадвалга яна бир устун кўшайлик. Унда ҳар бир номга мос изоҳ ёзилган бўлиб, фақат уяли қурилмалар учун кўринмасин. Кенг майдонли мониторларда, ҳусусан компьютер экранида ёки чоп этилаётганда кўриш мумкин бўлсин.

Жадвал яратишдан олдин унинг нечта устун ва сатрдан иборат эканлигини аниқлаб олиш мухим. Кўшма катаклар нечта яккасининг жойини эгаллашини кўрсатиш учун олдиндан жадвал тузилиши ҳақида тасаввур ҳосил қилиб олиш керак. Расмдан келиб чиқадиган бўлса, жадвал тўққизта устун ва йигирмата сатрдан иборат. Бунда ўширин изоҳли устун ва бош жойлар ҳам санаалган.

Жадвал мураккаб объект бўлиб, унинг ташки чегараси катакларнинг чегараларидан алоҳида чизилади. Шунинг учун битта умумий соя қўйиб бўлмайди. Чунки жадвал тўғри тўртбурчак шаклини эгаллайди, юқоридаги расмда эса қирралари ёйсимон тарзда.

Аввал ҳар бир катақка ўзига хос номдаги синф бериб HTMLда жадвални тузиб оламиз:

```
<table align="center"> /* жадвал майдоннинг ўртасига чиқади */
<tr> /* юқори ҳошия, бош ҳарфлар билан ёзилган */
    <td class="top upper" colspan="9" align="center">
        2-саҳифа
    </td>
</tr>
<tr class="bosh"> /* бўши қатор ва устунларни кўрсатади */
    /* Чап чегара: ингичка тусланувчан қора */
    <td class="left" rowspan="18"></td>
    <td class="yoni" rowspan="18"></td> /* Чап бўши устунлар */
    <td colspan="5"></td> /* Маълумот чиқувчи қисм: 5та устун */
    <td class="yoni" rowspan="18"></td> /* Ўнг бўши устунлар */
        /* Ўнг чегара: ингичка тусланувчан қора */
    <td class="right" rowspan="18"></td>
</tr>
<tr>
    <th class="title tl">№</th>
    <th class="title">Номи</th>
```

```

/* Яширик капиак */
<th class="title hideField">Күштимча маълумот</th>
<th class="title">Вазни<p class="note">(кг)</p></th>
<th class="title tr">Нархъ<p class="note">(сўм)</p></th>
</tr>
<tr> /* Кичик сарлавҳа */
  <th class="value" colspan="5">Мевалар</th>
</tr>
<tr>
  <td class="value" align="center">1.</td>
  <td class="value">Олма</td>
  <td class="value hideField">Олма изоҳи</td>
  <td class="value" align="center">2,4</td>
  <td class="value" align="right">7 000</td>
</tr>
<tr>
  <td class="value" align="center">2.</td>
  <td class="value">Нок</td>
  <td class="value hideField">Нок изоҳи</td>
  <td class="value" align="center">1,2</td>
  <td class="value" align="right">8 200</td>
</tr>
<tr>
  <td class="value" align="center">3.</td>
  <td class="value">Ўрик</td>
  <td class="value hideField">Ўрик изоҳи</td>
  <td class="value" align="center">3,6</td>
  <td class="value" align="right">9 300</td>
</tr>
<tr>
  <td class="value" align="center">4.</td>
  <td class="value">Беҳи</td>
  <td class="value hideField">Беҳи изоҳи</td>
  <td class="value" align="center">0,8</td>
  <td class="value" align="right">9 800</td>
</tr>
<tr>
  <td class="value" align="center">5.</td>
  <td class="value">Анор</td>
  <td class="value hideField">Анор изоҳи</td>
  <td class="value" align="center">1,3</td>
  <td class="value" align="right">19 300</td>
</tr>

```

```

<tr>
  <td class="value" align="center">6.</td>
  <td class="value">Шафтоли</td>
  <td class="value hideField">Шафтоли изохи</td>
  <td class="value" align="center">1,8</td>
  <td class="value" align="right">2 600</td>
</tr>
<tr>
  <td class="value" align="center">7.</td>
  <td class="value">Гилос</td>
  <td class="value hideField">Гилос изохи</td>
  <td class="value" align="center">2,7</td>
  <td class="value" align="right">3 700</td>
</tr>
<tr>
  <td class="title" align="right" colspan="2">Жами:</td>
  <td class="title hideField">&ampnbsp</td>
  <td class="title" align="center">13,8</td>
  <td class="title" align="right">59 900</td>
</tr>
<tr>
  <th class="value" colspan="5">Сабзавотлар</th>
</tr>
<tr>
  <td class="value" align="center">1.</td>
  <td class="value">Помидори</td>
  <td class="value hideField">Помидори изохи</td>
  <td class="value" align="center">3,6</td>
  <td class="value" align="right">9 300</td>
</tr>
<tr>
  <td class="value" align="center">2.</td>
  <td class="value">Бодринг</td>
  <td class="value hideField">Бодринг изохи</td>
  <td class="value" align="center">2,9</td>
  <td class="value" align="right">7 100</td>
</tr>
<tr>
  <td class="value" align="center">3.</td>
  <td class="value">Бақлажон</td>
  <td class="value hideField">Бақлажон изохи</td>
  <td class="value" align="center">1,8</td>

```

```

<td class="value" align="right">5 800</td>
</tr>
<tr> /* Кичик сарлағында td көтөгінде берилған */
<td class="title" align="right" colspan="2">>Жами:</td>
<td class="title \hideField">&nbsp;</td>
<td class="title" align="center">>8,3</td>
<td class="title" align="right">>22 200</td>
</tr>
<tr> /* Канімін сарлағында th көтөгінде берилған */
<th align="right" class="title bl upper" colspan="2">>
    Барчаси:
</th>
<th class="title \hideField">&nbsp;</th>
<th align="center" class="title">>22,1</th>
<th align="right" class="title br">>82 100</th>
</tr>
<tr class="bosh">
    <td colspan="4"></td>
</tr>
<tr> /* Пастки ҳошия */
    <td class="bottom" colspan="9" align="center">
        <div class="tugma"/* Эмлікс ичидеги міллар */>
            <a class="page" href="oldingi.html"
               title="Олдинги саҳифа">&lt;</a>
            <a class="page" href="keyingi.html"
               title="Кейинги саҳифа">&gt;</a>
        </div>
    </td>
</tr>
</table>

```

Энді ҳар бир «class»да күрсатылған номлар учун CSSда қоидалар мажмусини яратамиз. Юқоридаги расмдада фақат тұсланувчан қора (монохром) ранг танланған. Агар ташқы чегара ўзгармас аниқ битта ортраншта эга бўлганда, тасвир бу қадар ҳаққоний кўринмаган бўларди. Рангларни исталғанча ўзгартириш имкони мавжуд бўлиши учун SCSS тарзда [10.2.] қонуниятларни ёзамиз. Ўмумлаштирилган ҳолда чегаравий чизик кўриниши ва қирраси учун ёй радиусларини ҳам ўзгарувчилар орқали берамиз:

```

$black: #000; /* Қора ранг */
$white: #fff; /* Оқ ранг */
$dark: $black + #333; /* Тұқ  ранг */
$light: $white - #666; /* Оң ранг */
$line: 1px solid #aaa; /* Катаклар өзегараларининг күриниши */
$weight: 8px; /* Матндан өзегарағача ён масофа */
$border: 2px solid black; /* Ташиқи өзегаравий чизик */
$fontSize: 14px; /* Матн хусниҳатининг ўлчами */
$radius: 40px; /* Ташиқи өзегара құррасининг ёй радиуси */

@media handheld { /* фәқат смартфонларда чиқмаслиги үчүн */
    .hideField {
        visibility: hidden; /* күрінмаслиги үчүн */
        position: absolute; /* жой олмаслиги үчүн */
    }
}
table {
    border-spacing: 0; /* катаклар орасидаги масофа */
    border-collapse: separate; /* ҳар бир катакнинг үз өзегараси бор */
    margin: 10px; /* қўшини объектлардан ажраб түриши үчүн */
}
.top { /* Тена ташқи ҳосия */
    border: $border;
    border-bottom: none; /* Фәқат пастки өзегараси күрінмайды */
    border-top-left-radius: $radius;
    border-top-right-radius: $radius; /* түсланувчан қора */
    background-image: linear-gradient(to top, $dark, $light 65%);
    font-size: $fontSize - 2; /* юқоридаги матн ўлчами сал кичикроқ */
    padding: $fontSize + 2; /* катак баландроқ ва матн ўртада чиқади */
    color: $white; /* ортранған түқлілігі үчүн оқ рангда чиқади */
    box-shadow: 8px 0 10px $light;
} /* соя пастга тушмайды, фәқат ўнгдан чиқади */
.bottom { /* Пастки ташқи ҳосия */
    border: $border;
    border-top: none; /* Фәқат тена өзегара қүрінмайды */
    border-bottom-left-radius: $radius;
    border-bottom-right-radius: $radius; /* түсланиши */
    background-image: linear-gradient(to bottom, $dark, $light 75%);
    box-shadow: 6px 3px 10px 3px $light;
} /* Пастки өзегара бүйінча соя күрінади */

```

```

th.title { /* Кәмітә сарлағы */
    border: $line;
    padding: 2px $weight;
    font-size: $fontSize;
    color: $black;
    font-weight: bolder; /* Сәләр баландлығы көттәрең */
    line-height: $fontSize + $weight;
    vertical-align: middle; /* Матн үртага чиқиши үчүн */
    background-color: $white - #666; /* Орқаси түк күлранг */
}

th.value { /* Кичик сарлағы */
    font-size: $fontSize + 1;
    padding: 4px $weight;
    font-weight: bolder;
    border: $line;
    color: $dark; /* матн ранги түк */
    background-color: $white - #222; /* Орқаси салғина түккөрк */
}

td.title { /* Жами ёзиладиган кичик сарлағы */
    @extend th.value; /* Кичик сарлағы жиғолары билан бир хил */
    border-bottom-width: 5px; /* насткы қисмидә қалып чегара чизилади */
}

td.value { /* Қиймат чиқадиган катаң */
    font-size: $fontSize;
    padding: 2px $weight; /* Ёзув чегарадан ичкарирокда */
    border: $line;
    background-color: $white - #010101; /* Ортранги оқсифат */
}

/* Ички жадвал құрраларининг ёйи, 4 марта кичик */
.tl { border-top-left-radius: $radius / 4; }
.tr { border-top-right-radius: $radius / 4; }
.bl { border-bottom-left-radius: $radius / 4; }
.br { border-bottom-right-radius: $radius / 4; }

.left { /* Чап ташқы чегара */
    border-left: $border;
    width: 4px; /* Ортранги үртаси оч, чети түк түслегінде */
    background: radial-gradient(circle farthest-corner at center, $light, $dark);
}

```

```

.right {           /* Ўнг таисқи чегара */
    @extend .left;   /* Чап таисқи чегара сингари */
    border-left: none; /* фақат чап чегаравий чизик күрінмайды */
    border-right: $border; /* ўрнига ўнг томони күрінады */
    box-shadow: 7px 2px 10px 1px $light; /* ўнг томондан соя чиқады*/
}
.bosh { height: $weight; } /* Ички жадвалдан тепе-пастидаги масофа */
.yoni { width: $weight * 2; } /* ён чегарагача масофа, 2 мартта көттә */
.upper { text-transform: uppercase; } /* Болш ұғралар билан ёзилади */
.tugma { /* Эллипс шаклидаги түгма */
    border: 1px solid $black;
    border-radius: 16px;
    width: 32px;
    line-height: $fontSize * 2;
    margin: $weight;
    text-align: center;
    box-shadow: inset 3px 3px 5px $light;
}
/* Ажраб туриши учун соя ичкарига түшігандай */
.page { /* Эллипс ичидағы майлар шакли */
    font-weight: bold;
    font-family: Consolas;
    text-decoration: none; /* Ишорат матнининг тағ чизиги йүк */
    color: $black;
}
.note { /* Сарлағхадаги күшімчалар */
    font-style: italic; /* Әгерни ёзилғандай */
    color: $dark;
    font-size: $fontSize - 2; /* Үлчами кичикроқ */
    line-height: $fontSize - 6; /* Қисқа баландлықни әгалладайди */
    margin: 0; /* Тақи ҳошияси ұажм олмаслиғи учун */
}

```

10.9. Форма дарчалары мукаммал әмас!

Даставвал матнларни турли жилода расмлар билан намойиш этишга мүлжалланған браузер интернет тараққиеті ва оммалашиши натижасыда ўзига улкан вазифаларни ола бошлади. Эндилікда веб-саҳифалар

сервер билан мулоқот, маълумот алмасиши восита-си сифатида кенг хизмат кўрсатмоқда. Маълумотларни юбориш ва уларни қайта ишлаш учун HTMLдаги фор-ма дарчалари биз замонавий технологияларда ишлаб кўнинканимиздай имкониятларни тўлақонли бера ол-майди. Шунинг учун ҳам веб-интерфейс ясашда турли қўшимча кутубхоналар ёки дастурий мухитлардан фой-даланишга тўғри келади. Баъзи ҳолларда бунга сабаб HTML талабларига браузерларнинг жавоб бермаслиги (масалан FireFox ва Safariда вакт билан ишлаш ускуна-лари йўқ [7.3.]) бўлса, баъзан имкониятларининг ўзи ҳам мукаммал эмаслиги муаммо туғдиради. Масалан дарчага «readonly» аломати кўрсатилган бўлса, ундаги маълу-мотни ўзгартириш фойдаланувчи учун имконсиз. Аммо дарча тайинланувчи (select) бўлса, ундаги рўйхатдан бошқасини танлаб қийматини ўзгартиришга «readonly» қаршилик қила олмайди. Бунинг учун «disabled» ало-мати ишлатиладиган бўлса, бошқа дарчаларга нисбатан вазифаси жиҳатидан фарқ қиласди.

Бу тайинланувчи дарчанинг ягона камчилиги эмас. Агар унда бўш қийматли рўйхат бўлмаса, танлашни буткул бекор қилиш учун қўшимча скрипт ишлатиш-га мажбур бўлинади. Энг катта камчилиги, рўйхатнинг айни бир элементига «option» тегида ўзгача жило бе-риш мумкин, аммо у танланганда ёзуви қўринади, жи-лоси эса йўқолади. Амалий тажрибадан аён бўлдики, бунданда даҳшатли муаммоси мавжуд экан. Рўйхати келтирилган ҳар бир элемент учун ўзининг «option» тегида CSS ёрдамида алоҳида расм қўйилганда ва танлан-ган элемент қийматига мос равища «select»нинг ўзини ҳам расми ўзгарадиган қилинса, ушбу ажойиб тасвирили матнли тайинланадиган дарча фақат FireFoxда кўнгилда-гидек ишлар экан, энг оммабоп Chrome ҳам шу одий нарсани тўғри кўрсатиб бера олмас экан. Албатта HTML қоидаларида «select»нинг «option» элементларида расм қўйилиши ҳам қўлланилиши шарт дейилмаган, аммо умумий CSS имкониятлари бўйича бундай имконият мавжуд бўлиши кўзда тутилиши лозим эди.

Бу каби нуқсонларни кўплаб келтириш мумкин, шу-

нинг учун ҳам вебинтерфейс яратувчи мукаммалроқ ускуналарга эга дастурий мухитларни қўллаш тавсия этилади. Лекин ўрганишни улардан эмас, айнан ушбу китобда ёритилгандаидек HTML ва CSSнинг ўзагидан бошлиш зарур. Сабаби ундай мухитлар ўз қоидаларини ўрнатиб олиб, ўшандай ишлашга тақазо этади. Аслида эса у қоидалар ҳақиқий HTML ёки CSSда мавжуд бўлмаслиги, ўзгача мақсадда қўлланилиши ёки бутунлай бошқа маънони англатиши ҳам мумкин. Ўша мухитда мавжуд бўлмаган ёки қўллаш кўзда тутилмаган нарсаларни амалга оширишда албатта туб HTML, CSS ва JavaScript имкониятларини ишлатишга мажбур бўлинади.

Умуман олганда, ушбу китобда ёритилган билимлар ҳамда сўнги бўлимидағи келтирилган мисоллар ёрдамида HTML ва CSSлар орқали веб-саҳифада ихтиёрий кўринищдаги объектларни шакллантириш мумкин. Бу билан тўлақонли дизайнерлик фаолиятини ҳам юритса бўлади. Фақат озгина мусаввирлик салоҳияти ҳам мавжуд бўлса, фойдадан ҳоли эмас. Келгуси китобларимизда дастурловчи мақомига эга бўлиш учун билимлар ёритилади.

Теглар · күрсаткичи

!.....	13	dir	42
!DOCTYPE	14, 16	div.....	21, 142
a.....	48, 139	dl.....	46, 142
abbr	41, 139	dt	46, 142
acronym.....	41	em	39, 139
address	142	embed	42, 69
applet	42	fieldset.....	106
area	60	figcaption.....	70
article	23, 142	figure	70, 142
aside	23, 142	font	39, 44
audio	63, 142	footer.....	23, 142
b	38, 139	form.....	92, 142
base	18, 21	frame	42
basefont	39, 44	frameset	42
bdi	41, 139	h1	22, 142
bdo	40, 139	h2	22, 142
big.....	39, 44	h3	22, 142
blink.....	44	h4	22, 142
blockquote.....	142	h5	22, 142
body	16, 21	h6	22, 142
br	10, 37, 139	head.....	16, 17
button.....	100, 142	header	23, 142
canvas.....	73, 142	hgroup	23
caption.....	53	hr	142
center	42	html	17
cite.....	39, 139	i.....	39, 139
code.....	39, 139	iframe	70, 142
col	53	img	61, 63, 139
colgroup	54	input	94, 139
datalist.....	102	ins	39, 140
dd	46	isindex	42
del.....	39, 139	kbd.....	39, 140
details	47	keygen	103
dfn	39, 139	label.....	106, 140
dialog.....	188	legend.....	107

li.....	45, 46	s.....	39, 44, 140
link.....	18	samp.....	39, 140
listing.....	44	script.....	69
main	23	section.....	23, 143
map	50, 60	select.....	101, 140
mark	38, 140	small.....	39, 140
marquee	130	source.....	66
menu	43, 46	span.....	41, 140
meta	20, 21	strike.....	39, 44
meter.....	104, 140	strong	38, 140
nav.....	23, 142	style.....	33, 109
nextid	43	sub.....	39, 140
nobr	43	summary.....	47
noembed	43	sup.....	39, 140
noframes	43	svg	73
noscript	69	table	51, 143
object	43	tbody	54
ol.....	45, 142	td	55
optgroup.....	102	textarea	100, 140, 163
option	101	tfoot.....	54
output.....	105	th	55
p	38, 142	thead	54
param	107	time	40, 140
picture	67	title	21
plaintext	43	tr	55
pre	13, 43	tt	39, 44
progress.....	104	u	39, 44
q	39, 140	ul.....	42, 46
rb	44	var.....	39, 140
rp	140	video.....	64, 140
rt.....	140	wbr	139, 164
ruby	44, 140, 142	xmp.....	13, 43

CSS хусусиятлари

!important	109	border-left-color	147
align-content	159	border-left-style	147
align-items	159	border-left-width	147
align-self	159	border-radius	148
all	116	border-right	146
animation	130	border-right-color	147
animation-delay	131	border-right-style	147
animation-direction	132	border-right-width	147
animation-duration	130	border-spacing	147
animation-fill-mode	132	border-style	146
animation-iteration-count	132	border-top	146
animation-name	130	border-top-color	147
animation-play-state	132	border-top-left-radius	148
animation-timing-function	130	border-top-right-radius	148
backface-visibility	137	border-top-style	147
background	157	border-top-width	147
background-attachment	154	border-width	145
background-clip	154	bottom	150
background-color	154	box-shadow	145
background-image	122	box-sizing	144
background-origin	155	caption-side	162
background-position	156	clear	157
background-position-x	156	clip	159
background-position-y	156	color	164
background-repeat	156	column-count	160
background-size	155	column-fill	160
border	146	column-gap	160
border-bottom	146	column-rule	160
border-bottom-color	147	column-rule-color	160
border-bottom-left-radius	148	column-rule-style	160
border-bottom-right-radius	148	column-rule-width	160
border-bottom-style	147	columns	161
border-bottom-width	147	column-span	160
border-collapse	147	column-width	161
border-color	146	content	173
border-image	148	counter-increment	174
border-left	146	counter-reset	174

cursor	127
direction	161
display	140
empty-cells	163
filter	119
flex	153
flex-basis	153
flex-direction	152
flex-flow	153
flex-grow	153
flex-shrink	153
flex-wrap	152
float	157
font	166
font-family	165
font-kerning	165
font-size	165
font-stretch	165
font-style	165
font-variant	166
font-weight	166
height	144
hyphens	164
image-rendering	155
justify-content	159
left	150
letter-spacing	164
line-height	164
list-style	167
list-style-image	166
list-style-position	166
list-style-type	166
margin	144
margin-bottom	144
margin-left	144
margin-right	144
margin-top	144
marks	171
max-height	145
max-width	145
min-height	145
min-width	145
object-fit	161
opacity	167
order	153
orphans	172
outline	158
outline-color	158
outline-offset	158
outline-style	158
outline-width	158
overflow	161
overflow-x	162
overflow-y	162
padding	143
padding-bottom	143
padding-left	143
padding-right	143
padding-top	143
page-break-after	172
page-break-before	172
page-break-inside	172
pointer-events	173
position	150
quotes	163
resize	163
right	150
table-layout	163
tab-size	164
text-align	167
text-align-last	168
text-decoration	169
text-decoration-color	168
text-decoration-line	168
text-decoration-style	168
text-indent	169
text-overflow	169
text-shadow	169

text-transform	169	vertical-align	162
top	150	visibility	144
transform	137	white-space	170
transform-origin	136	widows	172
transform-style	137	width	144
transition	135	word-break	170
transition-delay	136	word-spacing	170
transition-duration	136	word-wrap	170
transition-property	135	writing-mode	171
transition-timing-function .	136	z-index	158
unicode-bidi	169	zoom	137
user-select	173		

ФОЙДАЛАНИЛГАН АДАБИЁТЛАР

- «Ахборот коммуникация технологиялари изоҳли луғати» Амиров Д.М. вб 2-н. Тошкент 2010
- «Веб-мастеринг: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка» Петр Ташков. «Питер» 2010
- «HTML и CSS. Путь к совершенству» Бен Хеник. «Питер» 2011
- «HTML5, CSS3 и Web 2.0. Разработка современных веб-сайтов» Дронов В. «БХВ-Петербург» 2011
- «HTML5. Рецепты программирования» Шмитт К. Симпсон К. «Питер» 2012
- «Пуленепробиваемый веб-дизайн. Библиотека специалиста» Седерхольм Д. З-н. «Питер» 2012
- «HTML и CSS. Разработка и дизайн веб-сайтов» Дакетт Джон. «Эксмо» 2013
- «Создай свой веб-сайт с помощью HTML и CSS» Йен Ллойд «Питер» 2013
- «Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5» Робин Никсон. 4-н. «Питер» 2013
- «Большая книга CSS3» Дэвид Сойер Макфарланд. «Питер» 2014
- «HTML5 и CSS3. Веб-разработка по стандартам нового поколения» Хоган Б. 2-н. «Питер» 2014
- «HTML5, CSS3 и JavaScript. Исчерпывающее руководство» Дженнифер Роббинс. «Эксмо» 2014
- «Современный веб-дизайн. Настольный и мобильный» Юлия Сырых. «Вильямс» 2014

- «CSS. Карманный справочник» Эрик А. Мейер. «Вильямс» 2016
- «Новая большая книга CSS» Дэвид Сойер Макфарланд. «Питер» 2016
- «Изучаем HTML, XHTML и CSS» Элизабет Фримен, Эрик Фримен. 2-н. «Питер» 2016
- «HTML5. Карманный справочник» Дженифер Роббинс. 5-н. «Вильямс» 2016
- «PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических Web-сайтов» Владимир Дронов. «БХВ-Петербург» 2016
- «HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств» Бен Фрейн. «Питер» 2017
- «<http://ilm.yurt.uz/>» веб-сайтинг кўллови учун алоҳида миннатдорчилик билдирамиз.

МУНДАРИЖА

1-бўлим. КИРИШ

1.1. Китоб ҳусусида	3
1.2. Қисқача тарих.....	4

2-бўлим. HTMLНИНГ АСОСИЙ ТАРКИБИ

2.1. Асосий тушунчалар.....	8
2.2. Илк босқич	13
2.3. Асосий қисмлар	15
2.4. Сахифанинг бош қисми.....	17
2.5. Сахифанинг асосий қисми	21
2.6. Аломатларга бериладиган қиймат турлари	25
2.7. Универсал аломатлар.....	29
2.8. Ҳодисалар	33
2.8.1. Сахифага боғлиқ ҳодисалар	34
2.8.2. Сичқонча ҳаракатига боғлиқ ҳодисалар:.....	34
2.8.3. Клавиатурага боғлиқ ҳодисалар:.....	35
2.8.4. Форма ҳодисалари	35
2.8.5. Медиа ҳодисалар:.....	35

3-бўлим. МАТННИ ЖОЙЛАШТИРИШ

3.1. Хатбоши ва блоклар.....	37
3.2. Матнни жилолаш	38
3.3. Эскирган теглар.....	41
3.4. Рўйхат.....	44
3.4.1. Саноқли рўйхат.....	45
3.4.2. Белгиланган рўйхат	45
3.4.3. Изоҳли рўйхат.	46
3.4.4. Изоҳи яширин рўйхат.....	47
3.5. Ишорат.....	48

4-бўлим. ЖАДВАЛЛАР

4.1. Жадвални эълон қилиш	51
4.2. Жадвалнинг сарлавҳаси	53

4.3. Жадвални шакллантириш	55
4.4. Жадвал чизиш	57
 5-бўлим. МЕДИА ФАЙЛЛАРНИ БОГЛАШ	
5.1. Харитали расм қўйиш	60
5.2. Товушли файл қўйиш	63
5.3. Лавҳали файл қўйиш	64
5.4. Танланган файлни юклаш	65
5.5. Ностандарт медиа қўйиш.....	69
5.6. Объектларни биректириш	70
5.7. Ташиб HTML файлни акслантириш	70
 6-бўлим. ГРАФИК ВА ТАСВИРЛАР ЧИЗИШ	
6.1. CANVAS теги.....	73
6.2. SVG теги.....	73
6.3. Геометрик шакллар қўйиш	77
6.3.1. Тўғри тўртбурчак	78
6.3.2. Доира	78
6.3.3. Эллипс.....	79
6.3.4. Кесма.....	79
6.3.5. Синиқ чизик.....	79
6.3.6. Кўпбурчак	80
6.4. Мураккаб шакллар чизиш.....	80
6.4.1. PATH теги	80
6.4.2. Маркердан фойдаланиш.....	84
6.4.3. Тасвирни ўзгача шакллантириш.....	85
6.4.4. Туслантириш.....	86
6.5. Ҳаракатланувчи тасвирлар	87
6.5.1. SET теги	87
6.5.2. ANIMATE теги.....	88
6.5.3. AnimateMotion теги	89
6.5.4. AnimateTransform теги	90
 7-бўлим. ФОРМА ЭЛЕМЕНТЛАРИ	
7.1. Форма яратиш	92

7.2. INPUT дарчаси	94
7.3. INPUT дарчаси турлари.....	97
7.4. Форма дарчалари	99
7.4.1. Тугма	100
7.4.2. Матн ёзиш дарчаси	100
7.4.3. Тайинланувчи (select) дарча.....	101
7.4.4. Танланувчи сатр	102
7.4.5. Маълумотларни шифрлаш.....	103
7.4.6. Жараёнларни кўрсатиш.....	104
7.5. Формага объектларни жойлаш	106

8-бўлим. CSS ТАРКИБИ

8.1. CSSни боғлаш.....	108
8.2. Умумий синтаксис.....	110
8.3. Қийматларнинг турлари	115
8.4. Функциялар	118
8.4.1. Ҳиралаштириш	119
8.4.2. Ёрқинлаштириш	119
8.4.3. Хижиллаштириш	120
8.4.4. Тўйинтириш.....	120
8.4.5. Шаффоффлик.....	120
8.4.6. Соя қўшиш.....	120
8.4.7. Оқ-қора тасвир	121
8.4.8. Расм рангини ўзгартириш.....	121
8.4.9. Орқатасвир ранглари.....	122
8.4.10. Ҳисобланган қиймат	124
8.5. Тањлов асосида бириктириш	124
8.6. Табақалаш.....	125
8.7. Ҳаракатлантириш	130
8.7.1. Ҳаракатлантирувчи хусусиятлар.....	130
8.7.2. Ҳаракатланишни таъминлаш	133
8.7.3. Хусусиятларларни умумлаштириш.....	134
8.7.4. Содда ҳаракатлантиришлар	135
8.7.5. Ҳаракатлантириш самаралари.....	136
8.8. Қаторли ва блокли теглар.....	138
8.9. Объект шакли	143

9-бўлим. ОБЪЕКТЛАРНИ ЖОЙЛАШ ВА ЖИЛОЛАШ

9.1. Объектни жойлаштириш.....	150
9.2. Орқа тасвир.....	153
9.3. Ташки элементларга нисбатан жойлаш	157
9.4. Ички элементларини акслантириш	159
9.5. Ҳос хусусиятлар	162
9.6. Ички матнни жилолаш.....	164
9.7. Чоп этиш хусусиятлари.....	171
9.8. Ҳодисаларни бошқариш.....	173
9.9. Вазияттан синфлаш	173

10-бўлим. МУКАММАЛ САЙТ ЯРАТИШ ТАЛАБЛАРИ

10.1. Бехато код ёзиш	177
10.2. Мослашувчан CSS яратиш	180
10.2.1. Ўзгарувчилар	182
10.2.2. Бириктириш	183
10.2.3. Файлдан юклаб олиш.....	183
10.2.4. Функциядан фойдаланиш	184
10.2.5. Мерос олиш	185
10.2.6. Математик амаллар	186
10.3. Модал ойна	188
10.4. Кисқа изоҳ	191
10.5. Ҳаракатланувчи меню.....	193
10.6. Қалқиб чиқувчи ойна	195
10.7. Йиғиладиган майдон	200
10.8. Мўъжаз ҳисботлар	203
10.9. Форма дарчалари мукаммал эмас!.....	210
Теглар кўрсаткичи	213
CSS хусусиятлари	215
Фойдаланилган адабиётлар	218

Илмий нашр

Жавлон АБДУЛЛО

МУКАММАЛ ДАСТУРЛАШ

1-китоб

HTML ва CSS

Мухаррир:
Амир ХУДОЙБЕРДИ

Техник мухаррир:
Равшанбек ТЕМИРОВ

Мусаҳҳих:
Элёр ДАРХОН

Дизайнер:
Маъруфжон РАҲМОНОВ

Нашриёт лицензияси AI № 126. 2008 йил 12 ноябрь
Теришга 2017 йил 10 августда берилди.

Босишига 2017 йил 26 сентябрда руҳсат этилди.
Бичими: 84x108 1/32. «Palatino Linotype» гарнитурасида
оффсет босма усулида оффсет қоғозида босилди.

14 шарт. б.т.
Адади: 500. 27-сон буюртма.
Баҳоси келишилган нархда.

Нашриёт манзили:
«Musiqा» нашриёти, Тошкент,
Олмазор кӯчаси, 1-уй.

“RENESSANS PRESS” МСНГ нинг матбаа бўлимида чоп этилди.
Манзил: Тошкент ш. Глинка, 21.